Fall 2019

Generalized Relay Network Design and Collaborative Dispatching in Truckload Transportation

Amin Ziaeifar

Southern Methodist University, aziaeifar@smu.edu

GENERALIZED RELAY NETWORK DESIGN AND
COLLABORATIVE DISPATCHING
IN TRUCKLOAD TRANSPORTATION

Approved by:

Dr. Halit Üster
Professor of EMIS
Dissertation Committee Chairperson

Dr. Sila Cetinkaya
Professor of EMIS

Dr. Canan Savaskan
Associate Professor of ITOM

Dr. Michael Hahsler
Assistant Professor of EMIS

Dr. Harsha Gangammanavar
Assistant Professor of EMIS

GENERALIZED RELAY NETWORK DESIGN AND COLLABORATIVE DISPATCHING IN TRUCKLOAD TRANSPORTATION

A Dissertation Presented to the Graduate Faculty of the

Lyle School of Engineering

Southern Methodist University

in

Partial Fulfillment of the Requirements

for the degree of

Doctor of Philosophy

with a

Major in Operations Research

by

Amin Ziaeifar

M.S., Industrial Engineering, University of Tehran
B.S., Industrial Engineering, Isfahan University of Technology

December 21, 2019

ACKNOWLEDGMENTS

I would like to express my deepest gratitude and appreciation to my advisor, Professor Halit Üster, for his guidance and encouragement. The completion of this dissertation would not have been possible without his insightful suggestions and unwavering support.

I am extremely grateful to my committee members Dr. Sila Cetinkaya, Dr. Canan Savaskan, Dr. Michael Hahsler, and Dr. Harsha Gangammanavar for serving as my committee members and their valuable advice and practical recommendations.

I would like to extend my deepest gratitude to my family because of their support, encouragement and patience throughout the years.

I also wish to thank the EMIS department faculties and staffs, my friends and officemates for their assistance and companions.

This research was supported by National Science Foundation (grant number CMMI 1538115).

Ziaeifar, Amin M.S., Industrial Engineering, University of Tehran, 2012
B.S., Industrial Engineering, Isfahan University of Technology, 2009

Generalized Relay Network Design and
Collaborative Dispatching
in Truckload Transportation

Advisor: Dr. Halit Üster

Doctor of Philosophy conferred December 21, 2019

Dissertation completed August 23, 2019

The truckload industry faces a serious problem of high driver shortage and turnover rate which is typically around 100%. Among the major causes of this problem are extended on-the-road times where drivers handle several truckload pickup and deliveries successively; non-regular schedules and get-home rates; and low utilization of drivers dedicated time. These are by-and-large consequences of the driver-to-load dispatching method, which is based on point-to-point dispatching or direct shipment from origin-to-destination, commonly employed in the industry. In this dissertation, we consider an alternative dispatching method that necessitates careful design of an underlying network. In this scheme, a truckload on its way to destination visits multiple relay nodes and the driver and/or tractor are switched with a new one at these locations so that each driver stays close to their home domicile. In this respect, we evaluate the project in three different parts in which we address strategic (long-term), tactical (medium-term) and operational (short-term) decisions to design, and examine the proposed network.

In the first part of this research, we study a tactical design of a relay point network (RP-network) that may potentially help to alleviate this problem. Some specific design characteristics include the possibility of both direct and RP-network shipments, multi-route assignments, fixed relay costs, limited route circuity, and coverage required for relay points. We present a MILP model capturing these characteristics and a solution procedure based on strengthened Benders decomposition framework further enhanced by efficient heuristics. The

solution approach is able to solve the large-scale problems, considering realistic inputs, in a reasonable time and helps us to examine the performance of the RP-network. Computational results demonstrate the performance of the algorithm.

In the second part, we investigate the strategic design of an RP-network under uncertainty in demand which can be more prominent for long-term planning. We use two-stage stochastic programming approach to model the RP-network designing problem in this situation. The setting of our problem of interest builds on the deterministic RP-network design problem addressed in the previous chapter. In this chapter, we extend this model by considering uncertainty in demands. The setting of our problem of interest builds on the deterministic RP-network design problem while we extend the model and the solution approach to address demand uncertainty. In order to address the computational difficulties specially occurring in this setting, we develop Progressive Hedging- Strengthened L-shaped algorithm. We show that the suggested solution method can effectively solve different classes of test instances and its effectiveness increases by increasing the size of the instances.

In the third part, we develop framework to study and test different truckload transportation concepts in an operational setting of our problem. This framework enables us to simulate day-to-day operations in TL transportation as closely as possible from the load dispatching and networking strategy perspectives. Using this simulation environment, we compare different network strategies including point-to-point (PtP), RP-network and hybrid PtP-RP-network and different dispatching approaches comprising dispatcher-based dispatching approach and a collaborative dispatching paradigm taking inputs from drivers as well. In this context, we develop and embed an optimization model for dispatching into our simulation environment.

Keywords: Relay network design, Benders decomposition, Two-stage stochastic programming, L-shaped method, Progressive hedging, Simulation-optimization

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

This dissertation is dedicated to my family,

especially my parents,

for their love, support, and encouragement.

Chapter 1

Introduction

1.1 Motivation

Trucking industry generated $700 billion in revenue in 2017, that is about 79% of the total freight revenue in the United States such that more than 70% of all domestic freight tonnage was transported by trucks for 10.77 billion tons of freight ([1]), and it is expected to rise to 20.73 billion tons in 2028 ([2]). The trucking industry employed 7.7 million people in 2017, where 3.5 million of them were drivers ([1]).

Two main types of truckload transportation in the United States are full-truckload (TL) and the less-than-truckload (LTL). In TL transportation, a full truckload is shipped directly from its origin to the destination by a single driver driving a single truck. This type of shipment is called point-to-point (PtP) dispatching. In order to minimize empty distances traveled, multiple loads are assigned to the driver in such a way that the final destination is close to his/her home base. Given a large geographical area as the US, these consecutive direct shipments, regardless of the difficulties of arrangement causing irregularities, usually create extended tour lengths leading to long away-from-home times for drivers. This issue convinces many TL drivers to quit and move to a job with a regular schedule ([3], [4]). This problem, called driver-turnover problem, is considered as a persistent issue for the US TL industry. By looking at the previously reported statistics it can be found that this industry has been dealing with this problem for a long time, as the turnover rate was reported 85%-110% in [5, 6] and 110%-120% in [7]. It still remains significantly high such that the annualized turnover rate at large truckload carriers in the first quarter of 2018 is about 94% ([8]), whereas its average in the last six quarters predating mid-2015 has been reported to be 93% ([9]). This high driver turnover rate incurs annual cost to the TL industry with 340,000 drivers around $2.8 billion reported in [10] and $3 billion reported in [11]. [10] reveal that the average of turnover cost per driver is about \$8234, and its range spans from \$2000 to \$21000. The main elements causing this cost are administrations, training, insurance, maintenance, and loss of profit ([12]). Offering attractive wages is a common approach to alleviate this problem, that in practice it is not really successful because of the nonmonetary roots discussed above.

The trucking industry also faces another critical issue, which is the driver shortage that has a tight connection with driver turnover. Based on the estimations, the driver shortage will grow from 50,000 drivers by the end of 2017 to more than 174,000 by 2026, if the current trends remain the same as it is reported in [13]. A discussion about different financial strategies, such as increasing wages and bonuses, to deal with the driver turnover and shortage issues in the trucking industry can be found in [14]. The persistence of the mentioned issues shows that these kind of approaches are not really effective. Given the ability of RP-network to reduce the away-from-home times for drivers, it can be considered as a viable alternative to PtP dispatching approach.

The rest of this dissertation is organized as follows: In Section 1.2 of this chapter, we provide a brief description of the RP-network. A review of the relevant literature is presented in Chapter 2 in which we cover the most important studies for all the works of this dissertation. The research objectives and approach are discussed in details in Chapter 3. In Chapter 4, we present our first study on modeling and algorithms on designing a general RP-network, and discuss our proposed solution procedure and related results. In Chapter 5, we study the RP-network design problem under demand uncertainty. We propose a two-stage stochastic programming model for the problem, suggest an effective solution approach and discuss the results. Finally, in Chapter 6 we discuss a simulation-based scheme demonstrating the operational settings of a truckload transportation network. Also, we provide a conclusion on all the works presented in Chapter 7.

1.2 Brief System Description

Relay points "RP"s, which are the main components of an RP-network, are the places where truckloads change drivers and/or tractors, and are different than hub locations which are intended to do sorting or consolidating of shipments. Being a domicile for a group of drivers and a service region - an area that can be covered by an RP - are two other functions that the RPs are supposed to have. Also, an RP can be considered as a domicile for a group of drivers and a service region for the area covered by it. Two types of drivers work in each RP-network, local and lane drivers. Local drivers are responsible to transfer the loads from the origins to region's RP (pick-ups) or from region's RP to the destination (deliveries). Lane drivers responsibility is to transfer loads between RPs. An RP-network is designed in such a way that driving distances, i.e., local and lane tour lengths, are controlled. In other words, there are predetermined maximum allowable distances for local and lane drivers to drive in each attempt. Moreover, utilizing an RP-network increases distances traveled by TLs in a circuitous basis. The difference between this circuitous path passing through the RPs and direct shipment can be restricted by utilizing percentage *circuity* constraints to control the delivery time and operational costs of the TLs.

Design and application of relay networks in TL transportation were previously studied in [15]. [16] evaluated the relay network design problem and its effect on the driver turnover rate in the TL transportation industry. More recently, [9] presented relay network design problem considering more general assumptions. [12] discussed the benefits of RP-networks other than regularizing the get-home rates for drivers. Improving truck utilization, generating more efficient schedules for trips and maintaining trained drivers leading to decreasing in training cost and having safer trips are among them. Also, reducing the parking space for fleets and decreasing its cost is another benefit of using RP-networks ([9]).

1.3 Research Objective and Approach

Figure 1.1 shows the workflow among different parts of the project. The strategic network design part or stochastic programming part determines the upper-level RPs locations as the long-term designing decision for the underlying RP-network. These locations are the input for the tactical designing or deterministic network design part. The outputs of this medium-term decision-making section, which are RPs locations and actual routing information, are considered as the inputs for the simulation model in RPs operations simulation part and will be used as the basis for load assignment procedure performed in this part. The ribbon at the left side of each part addresses the sequence of evaluating the mentioned sub-projects in this dissertation. First, we discuss the deterministic modeling part or P1, and then, the stochastic programming part or P2 is presented, and finally, we investigate the simulation model in part 3 or P3 to study driver to load assignment as the short-term decisions.

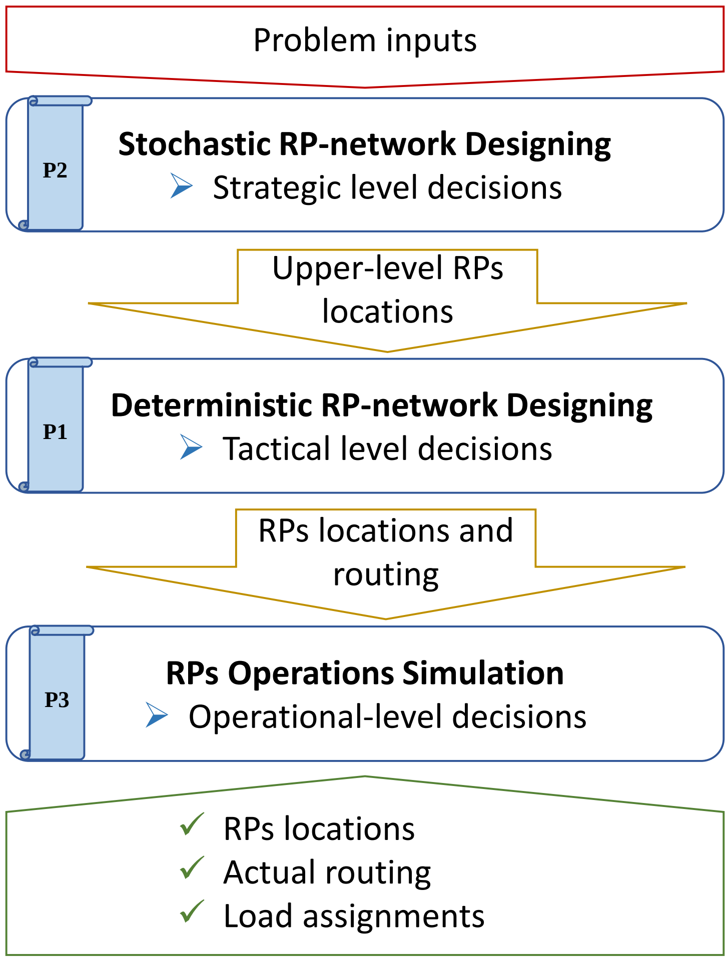

Figure 1.1: Overall workflow of the project

The main contributions of the first part of the project (P1), which is designing an RP-network under generalizing assumptions and deterministic demands, can be discussed as follows. The common assumption in all previous studies in the literature is the single assignment, which lets the origin/destination nodes be assigned to only one RP. In other words, a node that can potentially be an origin/destination of more than one commodity should be assigned to exactly one RP to receive or deliver associated demands. This assumption, although facilitates finding the solution for the problem, can increase the cost by increasing total distance traveled and decrease the generality of the problem. In this study, we relax this assumption in such a way that the nodes are assigned to RPs based on the requirements of every single commodity. This assumption, that we call it *multiple assignment*, makes it possible to assign a node to more than one RP. As mentioned above, using pure RP-networks for truckload shipments, although has benefits, can unnecessarily increase the total cost via introducing circuitous routes for the loads that can efficiently be transferred via direct shipment (PtP). This has been recognized before and addressed in an ad hoc fashion in [16] by choosing direct shipments loads in either before or after network design by considering direct distance and circuity levels, respectively. In this study, we consider both network and direct shipment simultaneously as possible ways of shipment. In this approach, it is possible to use PtP approach to ship those commodities for which utilizing RPs is not cost effective. Implementing this system, that we call *parallel shipment*, has specific requirements discussed later on in this study. It should be noted that using the direct shipment method causes increasing the total cost of the system implicitly because of its effect on increasing the driver turnover rate. We assume that this cost is embedded into the per unit direct shipment cost parameter value. In terms of the modeling, comparing to the model presented in [16], we introduce two new concepts, multiple assignment and parallel shipment, in the relay network design problem, making it more general and realistic, but challenging to solve. The circuity requirement in this new setting is another feature that we take into account in the proposed model.

In the context of the solution approach, a tabu search algorithm is presented by [15] to solve a relaxed version of the problem, by relaxing node imbalance and circuity constraints. [16] propose a Benders decomposition based approach to the problem considering node-imbalance, while they relax the percentage circuity constraint. Later, [9] present a Lagrangean decomposition based algorithm to solve a version of the problem considering capacity and link-imbalance constraints. In this study, we develop a strengthened Benders decomposition approach via a special type of disaggregated Benders cuts and two initial heuristics based on shortest path problem to warm start the Benders decomposition algorithm. The performance of the proposed algorithm is tested extensively in three different settings: with direct shipment and without circuity, with direct shipment and with circuity, and without direct shipment. We also perform a sensitivity analysis on the impact of the input parameters on the performance of the algorithm and design specifications of the proposed network.

In the second part, we investigate the strategic design of an RP-network under uncertainty in demand. This is important since having seasonal or fluctuating demand is inevitable in several industries and uncertainty in demand for long-term planning has to be handled explicitly. Expanding our finding in the first part of this project, we propose an exact solution approach to solve the problem optimally for a set of demand scenarios. We use two-stage stochastic programming approach to model the RP-network designing problem in this situation. The setting of our problem of interest builds on the deterministic RP-network design problem addressed in the previous chapter. In this chapter, we extend this model by considering uncertainty in demands. The setting of our problem of interest builds on the deterministic RP-network design problem while we extend the model and the solution approach to address demand uncertainty. In order to address the computational difficulties specially occurring in this setting, we develop a solution approach based on a new version of L-shaped method benefiting from initial information generated from a new version of Progressive Hedging algorithm. We call this approach Progressive Hedging-Strengthened L-shaped method and discuss its superiority in solving different classes of test instances.

The third part of this project focuses on evaluating the operations in the underlying network. In this part, we present a simulation framework in an operational setting of our problem to study alternative load-to-driver assignment and networking policies on our designed environment. The resulted models can simulate day-to-day operations in TL transportation as closely as possible from the load dispatching and networking strategy perspectives. Using this simulation environment, we can compare different network strategies including point-to-point (PtP), RP-network and hybrid PtP-RP-network and different dispatching approaches comprising dispatcher-based dispatching approach and a collaborative dispatching paradigm taking inputs from drivers as well. In this context, we develop and embed an optimization model for dispatching into our simulation environment. This optimization model tries to find the best settings for the underlying network to have the best results in terms of considered objective functions.

Chapter 2

Related Literature

Investigation in using relay points to shorten tour lengths and its capability in alleviating high driver turnover rate in the TL industry, has been started more two decades ago, when [17] present a simulation model to determine the design of the network and decide about using direct shipment as an alternative shipping approach. Suggesting simulation-based approaches were continued in [18], [19] and [20] where they evaluate the problem with different settings. [12] propose a simulation model taking percentage circuity and load imbalance considerations into account. The common suggestion of all the mentioned works above is using RP-networks to shorten tour lengths. They mention that well designing the network to optimize associated costs should be considered properly.

Utilizing non-simulation techniques to solve the problem was initiated in [21]. The author uses a shortest path based heuristic to design an RP-network where locating the RPs is not associated with fixed cost. Considering a similar setting, [22] present another shortest path based heuristic to design an RP-network while minimizing the number of RPs used. Mathematical formulation is not provided in these two studies. [15] formulate a mathematical model to design an RP-network capturing several generalizing assumption discussed earlier. They develop a tabu search algorithm to solve the model with relaxed load imbalance and percentage circuity constraints. Utilizing more exact algorithm to solve the problem is possible by improving the computers capabilities. An efficient Benders decomposition–based algorithm for an uncapacitated RP-network design model was recently presented by [16]. This paper investigates strength a well-designed RP-network to reduce driver turnover rate. The authors test the performance of the algorithm considering different assumptions including load imbalance and percentage circuity. Utilizing the same setting as [15] and [16], [23] present mixed-integer quadratic programming model including turnover cost in the objective to investigate driver-turnover problem. In terms of computational results to show the performance of the methodology, the authors suffice to a single node pair path results. Employing composite variables, generated by predefined routing pattern, [24] present an IP model to design a mixed fleet dispatching system for TL transportation combining direct and RP-network shipments. As the authors mention, even with a considered limitation on the number of RPs that can be at most visited in each route (the limitation is three RPs), the number of composite variables required for the formulation is extremely large. They propose a heuristic utilizing CPLEX to solve the problem. More recently, [9] introduce link capacity constraints and the concept of link imbalance in strategic relay network design, and solve a mixed-integer programming incorporating these concepts to design an RP-network. The network design problem with relays (NDR), a very similar problem to relay network design problem, is firstly introduced by [25] motivated by a telecommunication network design project. The authors present a path-based integer programming formulation and propose a column generation approach to obtain a lower bound. In this direction, [26] present multi-commodity flow and cut-set formulations for NDR problems with non-simple paths. Also, they present a branch-and-price and a branch-and-price-and-cut algorithm to solve this problem. [27] propose a multi-commodity flow and a tree formulation for NDR and NDR-S (single-source network design problem with relays) problems, respectively. Having a large number of variables associated with the arcs in their formulations, they propose branch-and-price algorithms to solve them.

In the context of modeling, configuration of our problem can be regarded as an extension of the single allocation hub location problem (HLP) ([28]). HLP tries to minimize the cost of locating hubs and uniquely allocating non-hub nodes to located hubs. HLP has a couple of specific assumptions including: each commodity should visit at least one hub in its route through the network, the subgraph induced by the hubs is complete and the commodities are consolidated among hub to hub shipments to reduce the total cost. Hence, a commodity visits at most two hubs on its path form the origin to the destination. The number of hubs (RPs) visited, in our problem, can be more than two, because of distance constraints discussed earlier. Also, considering parallel shipment, multiple assignment, percentage circuity and tour length assumptions make the problem more different than HLP. Review of the hub location problem can be found in [29], [30] and [31]. As we discussed, the RP-network design problem can categorized in the same category as incomplete HLP. Other than the works we introduced before, [32] define and formulate the single allocation incomplete p-hub median and the hub location with fixed costs network design problems and the single allocation incomplete hub covering and p-hub center network design problems. They solve these model via Cplex and analysis the impacts of changes in input parameters on the hub locations. Also, [33] present an integer programming formulation for the single allocation hub covering problem over incomplete hub networks and a tabu-based heuristic algorithm to solve it. [34] model the incomplete hub location problem with and without hop-constraints using a Leontief substitution system approach. The use Benders decomposition to the problem, while they devise a new scheme for generating Benders feasibility cuts to speed up the algorithm. [30] discuss several improving directions to be followed in HLP realm. The current work, similar to previous works presented by [15], [16] and [9], address some of the most important ones including employing incomplete network, having an overall cost perspective for multi-level hub-covering type network designs, considering realistic assumptions like circuity and proximity to model complexities in TL transportation. [30] also mention the limitations of the methodological contributions in the HLP literature which are mostly based on off-the-shelf solvers and heuristics. The solution approach devised in this paper, similar to those presented by [16] and [9], is able to be specialized to be utilized for an HLP.

In term of evaluating the RP-network problem in stochastic environment, this work is the first work. Before than this, as we discussed above, all the studies consider solving the problem in a deterministic environment. To review similar works, we evaluate a number of studies in which different types of network design problems are solved in the stochastic setting.

[35] develop a two-stage stochastic programming model for capacity-expansion problem considered in telecommunication network design setting. They suggest a L-shaped method utilizing a number of valid inequalities to solve the problem. [36] propose a two-stage robust optimization approach to solve network flow and design problems under demand uncertainty. Their computational study location-transportation problem reveals that their proposed approach suggests a trade-off between scenario-based stochastic programming and single-stage robust optimization which is more conservative.

More recently, [37] present a Benders decomposition algorithm to solve multi-commodity capacitated network design problem under demand uncertainty. They suggest a number of improving techniques for conventional Benders decomposition including use of stronger cuts, partial decomposition, heuristics, warm-start strategies, etc.

As we discuss later in Section 4.2.4, we propose a new Progressive Hedging (PH) algorithm for SMIP problems. Hence, we cover a number of studies in this area. PH was originally presented by [38] for linear stochastic programming models as powerful scenario decomposition method which convergence can be proved in a finite number of iterations. But its convergence for stochastic programming models with discrete decision variable scan not be proved. Hence, different versions of this method, as reliable heuristics, have been suggested to solve stochastic programming models with integer variables([39]; [40]). [41] suggest a number of algorithmic innovations to improve the PH algorithm in presence of integer variables. They test their algorithm on a class of scenario-based resource allocation problem and show that their suggestions are efficient to improve the PH convergence and runtime. [42] present a method to integrate PH and Dual Decomposition (DD) for stochastic mixed-integer programs. Their approach benefits from both approaches such that the converges of DD is accelerated by using the PH weights. A lower bounding technique for PH, to calculate the lower bound at any iteration of this algorithm, in solving two-stage and multi-stage stochastic mixed-integer programs is presented by [43]. They show that their bound is as tight as possible given the duality gap of integer programs.

PH have been applied to solve the network design and optimization problems. [44] use PH to create a solution method to solve their model for pre-disaster transportation network protection against uncertain future disasters, presented as a stochastic mixed-integer nonlinear program after a reformulation. They show that PH is effective in solving a broader range of applications consisting discrete and non-convex problems. [45] propose a metaheuristic algorithm, inspired by the progressive hedging algorithm, to solve a two-stage stochastic programming formulation proposed for the stochastic fixed-charge capacitated multicommodity network design (S-CMND) problem with uncertain demand. Their solution method outperforms a well-known commercial solver in terms of solution quality and computational effort. [46] present a methodological approach to devise strategies to group the underlying scenarios. They evaluate the suggested strategies through analyzing the performance of a new progressive hedging-based meta-heuristic solving subproblems, being made up of multiple scenarios, in the context of stochastic network design problem formulated as a stochastic mixed-integer program. They show that the PH-based meta-heuristic performs better when it solves multi-scenario subproblems compared to single-scenario.

In term of proposing simulation-based solution approach, as we mentioned earlier, there are several studies. We introduce a number of them as follows. [17] propose a simulation-based solution framework to evaluate hub-and-spoke transportation network to be used in truckload transportation operations. They develop a knowledge-based simulation model as a evaluation tool for their methodology. This work can be considered as the first work in the context of designing hub-and-spoke networks for truckload trucking. [47] present a simulation-based software system to evaluate Hub-and-Spoke transportation networks. They prepare a complete description of the software system, known as HUBNET including the motivating problem environment of truckload trucking, the overall simulation solution structure and the main aspects of the HUBNET system. [12] examine the use of multi-zone dispatching framework in truckload trucking using simulation methods and historical data. They suggest a new dispatching method offering compromise between the needs of the customer, the carrier, and the driver. This is the first study suggesting the use of multi-zone dispatching methods for truckload trucking.

Chapter 3

Relay Network Design with Direct Shipment and Multi-Relay Assignment (P1)

In this chapter we define the problem of designing an RP-network in the deterministic setting in detail, develop a mathematical model and propose a solution approach for the problem in a deterministic environment. The solution of the mentioned model provides the location of the RP nodes, assignment the origin/destination nodes to the RP nodes and actual routs to transport the loads. We present and discuss the computational results to show the validity of the model and performance of the solution algorithm.

3.1 Problem Definition and Formulation

In our formulation, we consider three sets with the following definitions: Set $\mathcal{N}$ represents the set of TL origin/destination nodes; Set $\mathcal{R}$ represents the set of potential RP locations; and set $\mathcal{Q}$ includes the commodities where a commodity $[i, j]$ is defined for a pair of nodes $i,\ j\ \in \mathcal{N}$ having a certain amount of demand (TLs) to be transported from origin i to destination j. Two methods of shipping commodities (truckloads) are possible. In the first one, a TL follows a path of comprised of only RPs and it can visit any RP in the first leg (as opposed to the uniquely determined RP that the origin node is assigned to as in [16]). Similarly, it can reach to its destination node in the last leg from any RP visited last in the path. That is, *multiple assignments* (as opposed to single assignments in which non-RP nodes are uniquely assigned to RP nodes) are possible for non-RP nodes. Alternatively, in the second one, a commodity can be shipped directly from its origin to its destination without using the RP-network. We call the first way of shipment "*network shipment*", the latter way, "*direct shipment*" and the overall system, "*parallel shipment*". Transportation between any two RPs is possible if the distance between them is not greater than Δ_2. Similarly, non-RP nodes can only send/receive TLs to/from the RPs within a distance of Δ_1. It should be noted that Δ_1 is typically less than Δ_2.

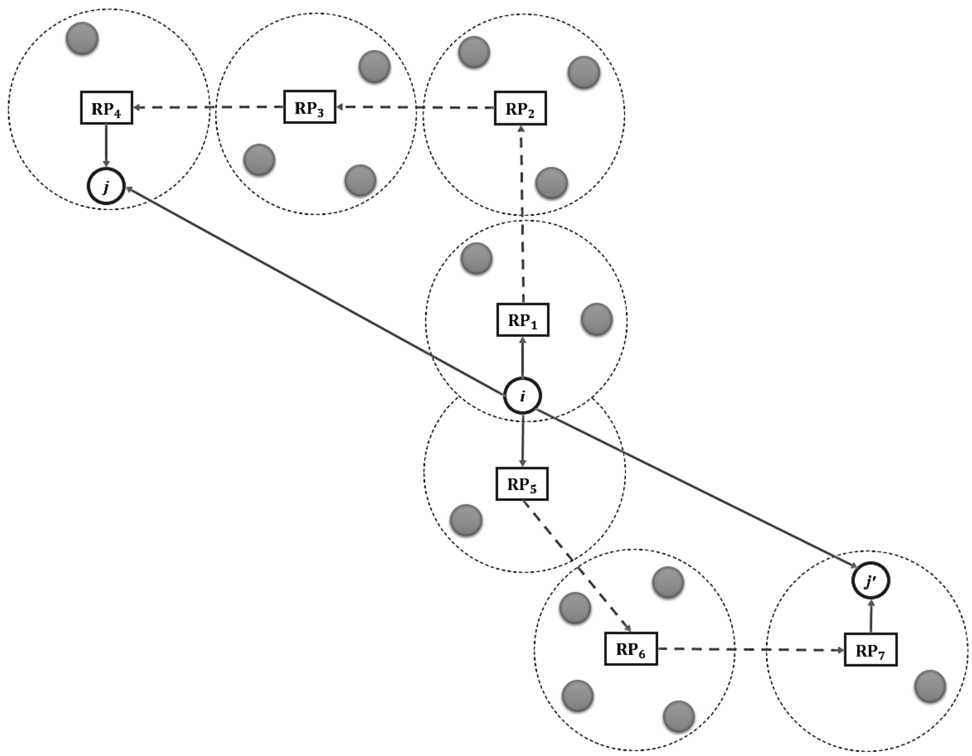

Figure 3.1: A Schematic View of an RP-Network

Figure 3.1 shows the operational characteristic of RP-network proposed in this study. Non-RP and RP nodes are represented by circles and squares, respectively. The bigger dashed circles represent the regions that RPs define. The radiuses of these circles are equal to allowable distance for local travels. Also, the distance between two squares is not greater than the allowable distance for lane transportation. Figure 3.1 also illustrates the assignments and possible routes to ship two different commodities associated with node i having destinations j and j'. The direct shipments choices are depicted by solid lines and network ones by dashed lines. As it is visible in this figure the node i is assigned to two different RPs to serve two different demands.

Using network shipment to ship the TLs associated with a commodity causes traveling a longer distance compared to direct shipment. In this study, we also employ percentage circuity constraints limiting additional distance traveled by trucks on a percentage basis for a commodity. The reason of using these constraints is that sometimes additional distances are not negligible and transportation companies prefer to limit them, although minimizing the transportation costs implicitly helps to optimize these additional distances.

The optimal solution of this problem determines the optimal location of the RPs, the assignment of non-RP nodes to RPs on an origin-destination node pair basis, and the optimal route for each commodity. The total objective function of the problem includes local and lane transportation costs of optimal routes associated with all commodities and fixed RP costs. The proposed model for this problem in Section 3.1.1 includes all these factors simultaneously.

3.1.1 Mathematical Model

We present the below notation followed by a formulation of our problem [MnP]:

Sets:

$\mathcal{N}$	set of commodity origin/destination nodes, $i,\ j \in \mathcal{N}$
$\mathcal{R}$	set of potential RP nodes, $k,\ l \in \mathcal{R}$
$\mathcal{Q}$	set of commodities, $[i,j] \in \mathcal{Q}$

Parameters:

w_{ij}	total demand for commodity $[i,j]$
d_{kl}	distance between node k and node l
t_1	transportation cost between RPs and non-RP nodes per-unit demand per-unit distance
t_2	transportation cost between two RPs per-unit demand per-unit distance
t_3	transportation cost between two non-RPs per-unit demand per-unit distance
f_k	fixed cost of locating an RP at node $k \in \mathcal{N}$
Δ_1	allowable distance between a non-RP node and an RP
Δ_2	allowable distance between two RP nodes
Ω	percentage circuity coefficient for allowable level of percentage circuity, $\Omega \geq 0$

Decision Variables:

x_{ij}^{kl}	1 if node i is assigned to the RP at node k and node j is assigned to the RP at node l

to ship commodity $[i,j] \in \mathcal{Q}$ using the RP network, $i,j \in \mathcal{N}$ and $k,l \in \mathcal{R}$

y_{ij}^{kl} 1 if commodity $[i,j] \in \mathcal{Q}$ uses the arc (k,l), $i,j \in \mathcal{N}$ and $k,l \in \mathcal{R}$,

z_{ij} 1 if commodity $[i,j] \in \mathcal{Q}$ shipped directly from node i to node j, $i,j \in \mathcal{N}$

h_k 1 if RP k, $k \in \mathcal{R}$ is used, 0 otherwise

$$\textbf{[MnP]} \qquad \text{Min} \quad \sum_k f_k h_k + \sum_i \sum_j \sum_k \sum_l t_1(d_{ik} + d_{jl}) w_{ij} x_{ij}^{kl} + \sum_i \sum_j \sum_k \sum_l t_2 d_{kl} w_{ij} y_{ij}^{kl} + \sum_i \sum_j t_3 d_{ij} w_{ij} z_{ij} \tag{3.1}$$

subject to

$$\sum_{\substack{m\in\mathcal{R}\\ m\neq k}} y_{ij}^{mk} + \sum_{\substack{m\in\mathcal{R}\\ m\neq k}} x_{ij}^{km} = \sum_{\substack{m\in\mathcal{R}\\ m\neq k}} y_{ij}^{km} + \sum_{\substack{m\in\mathcal{R}\\ m\neq k}} x_{ij}^{mk} \qquad \forall\, [i,j] \in \mathcal{Q},\ k \in \mathcal{R} \tag{3.2}$$

$$z_{ij} + \sum_{k\in\mathcal{R}} \sum_{l\in\mathcal{R}} x_{ij}^{kl} = 1 \qquad \forall\, [i,j] \in \mathcal{Q} \tag{3.3}$$

$$x_{ij}^{kl} \leq h_k \qquad \forall\, [i,j] \in \mathcal{Q},\ \forall\, k,\ l \in \mathcal{R} \tag{3.4}$$

$$y_{ij}^{kl} \leq h_l \qquad \forall\, [i,j] \in \mathcal{Q},\ \forall\, k,\ l \in \mathcal{R},\ \ k \neq l \tag{3.5}$$

$$\sum_{k\in\mathcal{R}} \sum_{l\in\mathcal{R}} (d_{ik} + d_{lj}) x_{ij}^{kl} + \sum_{k\in\mathcal{R}} \sum_{\substack{l\in\mathcal{R}\\ l\neq k}} d_{kl} y_{ij}^{kl} - d_{ij} \leq \Omega\, d_{ij} \qquad \forall\, [i,j] \in \mathcal{Q} \tag{3.6}$$

$$x_{ij}^{kl} \in \{0,1\} \qquad \forall\, [i,j] \in \mathcal{Q},\ \forall\, k,\ l \in \mathcal{R} \tag{3.7}$$

$$y_{ij}^{kl} \in \{0,1\} \qquad \forall\, [i,j] \in \mathcal{Q},\ \forall\, k,\ l \in \mathcal{R},\ \ k \neq l \tag{3.8}$$

$$z_{ij} \in \{0,1\} \qquad \forall\, [i,j] \in \mathcal{Q} \tag{3.9}$$

$$h_k \in \{0,1\} \qquad \forall\, k \in \mathcal{R} \tag{3.10}$$

The first part of the objective function measures the total fixed costs associated with locating the RPs. The second term represents the cost of total transportation between the origin/destination locations and RPs and the third one represents the total transportation cost between RPs, and finally, the fourth term is for direct shipment costs. Constraints

(3.2) are the flow conservation constraints for each node and commodity (TL). Constraints set (3.3) ensure that the demand should be sent through the relay network and/or direct shipment. Constraints set (3.4) and (3.5) regard locating the RPs. Constraints (3.6) are the percentage circuity constraints that control the weighted average of the additional travel distance incurred by using the RP network as opposed to direct shipment. The percentage of the weighted average extra travel distance must not be greater than $100 * \Omega$. Constraints (3.7) to (3.9) enforce the $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ to be binary. The model has the integrality property as we do not consider any capacity limitation on the links and RPs of the network and all the coefficients of the model are integral, these variables intrinsically only take values 0 or 1. Therefore, the three variables can also be stated as bounded continuous variables. In other words, the model can be solved as a mixed-integer program (MIP) instead of a pure integer program (IP). Constraints (3.10) are the binary requirements of the RP location variables $\mathbf{h}$.

3.2 Solution Methodology - Strengthened Benders Decomposition

Benders decomposition (BD) method decomposes the overall MIP formulation into a master problem and a subproblem, and solves them iteratively while exchanging their solutions. The master problem includes all the integer variables of the problem and associated constraints. Also, it contains an auxiliary continuous variable facilitating the interaction between the master and subproblem. The subproblem, on the other hand, is a linear program containing all the continuous variables of the MIP and associated constraints incorporating the integer variables as fixed values coming from the solution of the master problem. The master problem provides a lower bound for a minimization problem and an upper bound is attained by solving the (dual of the) subproblem. The bounds are updated in each iteration and the procedure is terminated as the difference between these two bounds is less than a predetermined negligible value. The Benders cuts are generated and added to the master problem during the implementation of the algorithm until it reaches the stopping criterion. We next discuss the details of the overall approach and the specific tools that we suggest for enhancing its performance.

3.2.1 Benders Subproblem (PBSP) and its Dual

As we discussed above, the model has integrality property, hence we formulate the PBSP as a linear program. For the given fixed RP locations $\hat{h}_k$, $k \in \mathcal{R}$, we can write PBSP as the following linear program.

$$[\mathbf{PBSP}] \quad \text{Min} \quad \sum_i \sum_j \sum_k \sum_l t_1(d_{ik} + d_{jl}) w_{ij} x_{ij}^{kl} + \sum_i \sum_j \sum_k \sum_l t_2 d_{kl} w_{ij} y_{ij}^{kl} + \sum_i \sum_j t_3 d_{ij} w_{ij} z_{ij} \tag{3.11}$$

subject to

$$\sum_{\substack{m\in\mathcal{R}\\ m\neq k}} y_{ij}^{mk} + \sum_{\substack{m\in\mathcal{R}\\ m\neq k}} x_{ij}^{km} = \sum_{\substack{m\in\mathcal{R}\\ m\neq k}} y_{ij}^{km} + \sum_{\substack{m\in\mathcal{R}\\ m\neq k}} x_{ij}^{mk} \qquad \forall\, [i,j] \in \mathcal{Q},\ k \in \mathcal{R} \tag{3.12}$$

$$z_{ij} + \sum_{k\in\mathcal{R}} \sum_{l\in\mathcal{R}} x_{ij}^{kl} = 1 \qquad \forall\, [i,j] \in \mathcal{Q} \tag{3.13}$$

$$x_{ij}^{kl} \leq \hat{h}_k \qquad \forall\, [i,j] \in \mathcal{Q}, \forall\, k,\ l \in \mathcal{R} \tag{3.14}$$

$$y_{ij}^{kl} \leq \hat{h}_l \qquad \forall\, [i,j] \in \mathcal{Q}, \forall\, k,\ l \in \mathcal{R},\ \ k \neq l \tag{3.15}$$

$$\sum_{k\in\mathcal{R}} \sum_{l\in\mathcal{R}} (d_{ik} + d_{lj}) x_{ij}^{kl} + \sum_{k\in\mathcal{R}} \sum_{\substack{l\in\mathcal{R}\\ l\neq k}} d_{kl} y_{ij}^{kl} - d_{ij} \leq \Omega\, d_{ij} \qquad \forall\, [i,j] \in \mathcal{Q} \tag{3.16}$$

$$x_{ij}^{kl} \geq 0 \qquad \forall\, [i,j] \in \mathcal{Q},\ \forall\, k,\ l \in \mathcal{R} \tag{3.17}$$

$$y_{ij}^{kl} \geq 0 \qquad \forall\, [i,j] \in \mathcal{Q},\ \forall\, k,\ l \in \mathcal{R},\ \ k \neq l \tag{3.18}$$

$$z_{ij} \geq 0 \qquad \forall\, [i,j] \in \mathcal{Q} \tag{3.19}$$

Note that the upper bounds for the variables of PBSP (which are 1 for all of them) are implicitly enforced by the Constraint (3.12) and (3.13). To obtain the dual of the PBSP, the variables ${q_1}_{ij}^{k}$, q_{2ij}, ${q_3}_{ij}^{kl}$, ${q_4}_{ij}^{kl}$, and q_{5ij} are defined as the duals of the constraints (3.12), (3.13), (3.14), (3.15), and (3.16), respectively. Then, we have

$$[\mathbf{DBSP}] \quad \text{Max} \quad \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{k\in\mathcal{R}}\sum_{l\in\mathcal{R}}(\hat{h}_k {q_3}_{ij}^{kl} + \hat{h}_l {q_4}_{ij}^{kl}) + \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}}({q_2}_{ij} + (1+\Omega)d_{ij}{q_5}_{ij}) \tag{3.20}$$

subject to

$$\begin{aligned}
&{q_1}_{ij}^{k} - {q_1}_{ij}^{l} + {q_2}_{ij} + {q_3}_{ij}^{kl} + (d_{ik}+d_{lj}){q_5}_{ij} \\
&\qquad\qquad \le t_1(d_{ik}+d_{lj})w_{ij} && \forall\, [i,j] \in \mathcal{Q},\ \forall\, k,\ l \in \mathcal{R} && (3.21)\\
&{q_1}_{ij}^{l} - {q_1}_{ij}^{k} + {q_4}_{ij}^{kl} + d_{kl}{q_5}_{ij} \le t_2 d_{kl} w_{ij} && \forall\, [i,j] \in \mathcal{Q},\ \forall\, k,\ l \in \mathcal{R},\ k \ne l && (3.22)\\
&{q_2}_{ij} \le t_3 d_{ij} w_{ij} && \forall\, [i,j] \in \mathcal{Q} && (3.23)\\
&{q_3}_{ij}^{kl},\ {q_5}_{ij} \le 0 && \forall\, [i,j] \in \mathcal{Q},\ \forall\, k,\ l \in \mathcal{R} && (3.24)\\
&{q_4}_{ij}^{kl} \le 0, && \forall\, [i,j] \in \mathcal{Q},\ \forall\, k,\ l \in \mathcal{R},\ \ k \ne l. && (3.25)
\end{aligned}$$

Let $\mathcal{E}$ denote the set of all extreme points of the DBSP polyhedron given by (3.21)-(3.25) and ${q_2}_{ij}{}^{e},\ {q_3}_{ij}^{kl^e},\ {q_4}_{ij}^{kl^e},\ {q_5}_{ij}{}^{e}$ and η^e denote the associated dual variables and objective function value with extreme point $e \in \mathcal{E}$. Letting η^* be the optimal objective value for the portion of original MIP's objective value employed in PBSP, we must have $\eta^* \ge \eta^e$, $\forall e \in \mathcal{E}$, and, thus, DBSP can be restated as $\min_{\eta\ge 0}\{\eta : \eta^e \le \eta,\ \forall e \in \mathcal{E}\}$ where

$$\eta^e = \sum_i\sum_j\sum_k\sum_l(\hat{h}_k {q_3}_{ij}^{kl^e} + \hat{h}_l {q_4}_{ij}^{kl^e}) + \sum_i\sum_j({q_2}_{ij}{}^{e} + (1+\Omega)d_{ij}{q_5}_{ij}{}^{e}) \tag{3.26}$$

3.2.2 Reformulation and the Benders Master Problem (BMP)

Utilizing the above representation of the DBSP that is based on the extreme points of its polyhedron, we can reformulate our model MnP as

$$\text{Min} \quad \sum_k f_k h_k + \eta \tag{3.27}$$

subject to

$$\eta \geq \sum_i \sum_j \sum_k \sum_l (h_k {q_3}_{ij}^{kl^e} + h_l {q_4}_{ij}^{kl^e}) + \sum_i \sum_j ({q_2}_{ij}{}^e + (1+\Omega) d_{ij} {q_5}_{ij}{}^e) \qquad \forall\, e \in \mathcal{E} \tag{3.28}$$

$$h_k \in \{0,1\} \qquad \forall\, k \in \mathcal{R}. \tag{3.29}$$

It should be noted that, in the BD solution framework, optimality cuts (3.28) are generated iteratively since all the constraints in (3.28) are not binding in the optimality. Hence, the reformulation containing only a subset of these constraints at any iteration t is a relaxation for BMP and its optimal solution provides a lower bound on the MnP. We can also include (3.30), called feasibility cut,

$$\sum_i \sum_j \sum_k \sum_l (h_k {q_3}_{ij}^{kl^{e'}} + h_l {q_4}_{ij}^{kl^{e'}}) + \sum_i \sum_j {q_2}_{ij}{}^{e'} \leq 0 \qquad \forall e' \in \mathcal{E}' \tag{3.30}$$

to this reformulation to avoid the extreme rays of the DBSP feasible region when a solution of BMP cannot guarantee feasibility of the BPSP. The cuts (3.30) may essentially be needed in the case that the direct shipment is not allowed and the network shipment is the only way of shipping the truckloads. We discuss this case, as one of the cases evaluated in this study, in Section 3.2.6. Note that the feasibility cut (3.30) does not contain the term $(1+\Omega)d_{ij}{q_5}_{ij}{}^{e'}$ since the circuity constraint is not addressed in this case.

3.2.3 Strengthening the Benders' Cuts

Notice that the subproblem has network flow problem structure dealing with potential degeneracy. Therefore, the DBSP may have multiple optimal solutions, each of which provides a different cut with different strength in terms of cutting off the BMP feasible area. This is important since by picking a proper optimal solution of DBSP in each iteration of

Benders' decomposition algorithm, we can generate stronger Benders' cut and accelerate the algorithm. For an optimization problem $Min_{y\in Y, z\in R}\{z:\ f(u)+y\,g(u)\leq z,\ \forall u\in U\}$, [48] define the concept of strongness of a cut as follows. The cut $f(u_0)+y\,g(u_0)\leq z$ is stronger than the cut $f(u_1)+y\,g(u_1)\leq z$ if $f(u_0)+y\,g(u_0)\geq f(u_1)+y\,g(u_1)$ $\forall y\in Y$ with a strict inequality for at least one $y\in Y$. In order to generate a strengthened Benders cut, we use a technique similar to the two-phase approach presented by [49] and also adapted by [50]. For our problem, we observe that, at an iteration of the algorithm, the dual values associated with $\hat{h}_k$ with a zero value do not have any contribution in the optimal objective value of DBSP. Hence, it is possible to modify the coefficient of a $\hat{h}_k=0$ ($\hat{h}_l=0$), that is ${q_3}_{ij}^{kl}$ (${q_4}_{ij}^{kl}$), without any effect on the DBSP objective function value. To do so, we first solve the reduced DBSP to attain the dual variable values (${q_3}_{ij}^{kl}$ and ${q_4}_{ij}^{kl}$) associated with open RPs ($\hat{h}_k=1$ and $\hat{h}_l=1$), and the optimal values for ${q_1}_{ij}^{k}$, ${q_2}_{ij}$, ${q_2}_{ij}$ and ${q_5}_{ij}$ $\forall\,[i,j]\in\mathcal{Q}$, $\forall\,k\in\mathcal{R}$ as well. To generate this problem, we define $\mathcal{G}$ as the set of open RPs at an iteration. Afterwards, we can have the mentioned values by solving the following problem. Specifically, we have

$$[\mathit{Reduced}\ \mathbf{DBSP}]\quad \text{Max}\quad \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{k\in\mathcal{G}}\sum_{l\in\mathcal{G}}(\hat{h}_k{q_3}_{ij}^{kl}+\hat{h}_l{q_4}_{ij}^{kl}) + \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}}({q_2}_{ij}+(1+\Omega)d_{ij}{q_5}_{ij}) \tag{3.31}$$

subject to

$$\begin{aligned}
&{q_1}_{ij}^{k}-{q_1}_{ij}^{l}+{q_2}_{ij}+{q_3}_{ij}^{kl}+(d_{ik}+d_{lj}){q_5}_{ij} \\
&\qquad\qquad\leq t_1(d_{ik}+d_{lj})w_{ij} && \forall\,[i,j]\in\mathcal{Q},\ \forall\,k,\ l\in\mathcal{G} && (3.32)\\
&{q_1}_{ij}^{l}-{q_1}_{ij}^{k}+{q_4}_{ij}^{kl}+d_{kl}{q_5}_{ij}\leq t_2d_{kl}w_{ij}, && \forall\,[i,j]\in\mathcal{Q},\ \forall\,k,\ l\in\mathcal{G},\ k\neq l && (3.33)\\
&{q_2}_{ij}+(1+\Omega)d_{ij}{q_5}_{ij}\leq t_3d_{ij}w_{ij}, && \forall\,[i,j]\in\mathcal{Q} && (3.34)\\
&{q_3}_{ij}^{kl},\ {q_5}_{ij}\leq 0 && \forall\,[i,j]\in\mathcal{Q},\ \forall\,k,\ l\in\mathcal{R} && (3.35)\\
&{q_4}_{ij}^{kl},\leq 0, && \forall\,[i,j]\in\mathcal{Q},\ \forall\,k,\ l\in\mathcal{G},\ k\neq l. && (3.36)
\end{aligned}$$

and letting $\mathcal{G}' = \mathcal{R} \backslash \mathcal{G}$, we solve the following model for a set of dual variable values, that are associated with $\hat{h}_k = 0,\ \hat{h}_l = 0\ \forall k, l \in \mathcal{G}'$, to generate a strengthened Benders cut.

$$\text{Max} \quad \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} \sum_{k \in \mathcal{G}'} \sum_{l \in \mathcal{G}'} ({q_3}_{ij}^{kl} + {q_4}_{ij}^{kl}) \tag{3.37}$$

subject to

$$\hat{q_1}_{ij}^{k} - \hat{q_1}_{ij}^{l} + \hat{q}_{2ij} + {q_3}_{ij}^{kl} + (d_{ik} + d_{lj})\hat{q}_{5ij}$$
$$\leq t_1(d_{ik} + d_{lj})w_{ij} \qquad \forall\, [i,j] \in \mathcal{Q},\ \forall\, k,\ l \in \mathcal{G}' \tag{3.38}$$

$$\hat{q_1}_{ij}^{l} - \hat{q_1}_{ij}^{k} + {q_4}_{ij}^{kl} + d_{kl}\hat{q}_{5ij} \leq t_2 d_{kl} w_{ij}, \qquad \forall\, [i,j] \in \mathcal{Q},\ \forall\, k,\ l \in \mathcal{G}',\ k \neq l \tag{3.39}$$

$${q_3}_{ij}^{kl} \leq 0, \qquad \forall\, [i,j] \in \mathcal{Q},\ \forall\, k,\ l \in \mathcal{G}',\ k \neq l \tag{3.40}$$

$${q_4}_{ij}^{kl} \leq 0, \qquad \forall\, [i,j] \in \mathcal{Q},\ \forall\, k,\ l \in \mathcal{G}',\ k \neq l. \tag{3.41}$$

The optimal solution of the (3.37)-(3.41) is attainable using the following approach without actually solving a linear program. We can rewrite (3.38) as ${q_3}_{ij}^{kl} \leq \delta_{ij}^{kl}$ where $\delta_{ij}^{kl} = t_1(d_{ik} + d_{lj})w_{ij} - (\hat{q_1}_{ij}^{k} - \hat{q_1}_{ij}^{l} + \hat{q}_{2ij} + (d_{ik} + d_{lj})\hat{q}_{5ij}),\ \forall\, [i,j] \in \mathcal{Q},\ \forall\, k,\ l \in \mathcal{G}'$. Thus, the optimal value of ${q_3}_{ij}^{kl}$ can found using the following equation

$${q_3}_{ij}^{kl} = \begin{cases} \delta_{ij}^{kl} & \text{if } \delta_{ij}^{kl} < 0 \\ 0 & \text{otherwise} \end{cases} \quad \forall\, [i,j] \in \mathcal{Q},\ \forall\, k,\ l \in \mathcal{G}'.$$

The optimal value of ${q_4}_{ij}^{kl}$ is obtained similarly by utilizing (3.39).

3.2.4 Surrogate Constraints for BMP

When the direct shipment is not an option in the problem, the feasibility of the PBSP for each solution of BMP cannot be guaranteed. Therefore, the feasibility cuts based on extreme rays of BDSP are required. Having several feasibility cuts can hinder the efficiency of the approach as the runtime to solve the BMP becomes excessive. To avoid this issue, we introduce a set of surrogate constraints in the master problem in an attempt to decrease the number of feasibility cuts required dramatically. These cuts can also assist to improve the lower bound of the problem by creating an effective network structure as BMP solution. To this end, we first define an auxiliary integer variable as

v_{kl} amount of flow of all the commodities on the arc (k, l), $k, l \in \mathcal{N} \cup \mathcal{R}$

Notice that, we do not distinguish commodities that make up the flow on arc (k, l), we state v_{kl} as the aggregate flow amount. This is because we really do not need these flow values as an output of the BMP, but rather need them to help with setting up RPs in good locations to facilitate feasibility of the PBSP. Utilizing v_{kl}, we define the following valid, surrogate constraints to be added to the master problem. These constraints help to establish relay point locations (item 3 below) in such a way that the subproblem has a feasible solution and optimality cuts are generated based on the solution of its dual problem DBSP.

1. Constraint sets (3.42) and (3.43) ensure that at least one arc is open to send and receive the flow of commodities for each origin and destination node, respectively.

$$\sum_{j \in \mathcal{N}_\mathcal{D}} w_{ij} \le \sum_{l \in \mathcal{R}} v_{il} \qquad \forall\, i \in \mathcal{N}_\mathcal{O} \tag{3.42}$$

$$\sum_{i \in \mathcal{N}_\mathcal{O}} w_{ij} \le \sum_{k \in \mathcal{R}} v_{kj} \qquad \forall\, j \in \mathcal{N}_\mathcal{D} \tag{3.43}$$

   where $\mathcal{N}_\mathcal{O}$ and $\mathcal{N}_\mathcal{D}$ are the sets of origin and destination nodes, respectively. The validity of the constraint (3.42) follows from the fact that some arcs connecting a source node to some relay node(s) must be used for all TLs to get on the RP-induced network; Similarly by (3.43) for all the TLs to arrive in their destinations.

2. Constraint sets (3.44) and (3.45) are essentially the flow conservation constraints expressed as two inequalities for each RP location.

$$\sum_{k \in \mathcal{R} \cup \mathcal{N}_\mathcal{O}} v_{kl} \le \sum_{m \in \mathcal{R} \cup \mathcal{N}_\mathcal{D}} v_{lm} \qquad \forall\, l \in \mathcal{R} \tag{3.44}$$

$$\sum_{m \in \mathcal{R} \cup \mathcal{N}_\mathcal{D}} v_{lm} \le \sum_{k \in \mathcal{R} \cup \mathcal{N}_\mathcal{O}} v_{kl} \qquad \forall\, l \in \mathcal{R} \tag{3.45}$$

3. Finally, constraint set (3.46) contains a set of linking constraints for activating RP

candidates as open locations.

$$v_{kl} \leq M\, h_l \qquad\qquad \forall\, k \in \mathcal{N} \cup \mathcal{R},\ \forall\, l \in \mathcal{R} \tag{3.46}$$

Having an open arc originates from or destines to an RP, we should have that RP active. These constraints enforce opening required RPs and are valid. To ensure having a tight and model, we should consider the M value properly. Since this value can be at most the total demand in the system, $\sum_{i \in \mathcal{N_O}} \sum_{j \in \mathcal{N_D}} w_{ij}$ can be regarded as a reliable value for M.

The proposed surrogate constraints perform very well in terms of recreating the network structure for the master problem. Although they can not guarantee the feasibility of the subproblem, they can keep the number of required feasibility cuts small and decrease the runtime dramatically. We discuss the performance of these surrogate constraint in Section 3.3.3.3.

We discuss another set of surrogate constraints, called shortest path cuts, later in Section 3.2.7 where we introduce two initializing heuristics for the Bender Decomposition algorithm

3.2.5 Cut Disaggregation Schemes

Observe that the subproblem PBSP is separable for each commodity $[i, j]$ giving $|\mathcal{Q}|$ independent routing problems over the network including origin and destination nodes in addtion to the RP nodes established by the BMP solution. Based on this observation, we introduce four different cut disaggregation schemes that can help to improve the performance of conventional single Benders cut (3.47) of the form (3.28).

$$\eta \geq \sum_i \sum_j \sum_k \sum_l (h_k {q_3}_{ij}^{kl} + h_l {q_4}_{ij}^{kl}) + \sum_i \sum_j ({q_2}_{ij} + (1+\Omega) d_{ij} {q_5}_{ij}) \tag{3.47}$$

Note that all alternative Benders cuts below also follow the same form which is based on DBSP objective function (3.20).

OD Cut The Benders cut can be separated based on every single commodity so that one cut is added for each commodity. Thus, in each iteration $|\mathcal{Q}|$ cuts would be added to the BMP. These cuts are

$$\eta_{ij} \geq \sum_{k\in\mathcal{R}}\sum_{l\in\mathcal{R}}(h_k {q_3}_{ij}^{kl} + h_l {q_4}_{ij}^{kl}) + q_{2ij} + (1+\Omega)d_{ij}q_{5ij} \qquad \forall\ [i,j]\ \in \mathcal{Q}.$$

O Cut The second cut type addresses disaggregation based on commodities' origin nodes such that we add one Benders cut for each node which is origin of at least one commodity. Thus, Benders cuts for all commodities originated at the same node are aggregated into one cut and there exists one such cut for each origin node. Defining set $\mathcal{Q}_O$ as the set of origin nodes, these cuts are

$$\eta_i \geq \sum_{\{j|[i,j]\in\mathcal{Q}\}}\sum_{k\in\mathcal{R}}\sum_{l\in\mathcal{R}}(h_k {q_3}_{ij}^{kl} + h_l {q_4}_{ij}^{kl}) + \sum_{\{j|[i,j]\in\mathcal{Q}\}}(q_{2ij} + (1+\Omega)d_{ij}q_{5ij}) \qquad \forall\ i \in \mathcal{Q}_O.$$

It should be noted that the performance of this cut and disaggregation based on commodities' destinations is almost same and that this type of cut disaggregation was first suggested by [16].

ODReg Cut In addition to commodity (origin-destination nodes) based disaggregation as in OD Cut above, disaggregation can also be done based on a regional basis for origin-destination pairs.

To achieve this, we divide the geographical area by a number of equally sized rectangles (regions) and generate one cut for commodities originating in and destined to the same regions. In other words, we aggregate the cuts of commodities based on their origin and destination regions. Figure 3.2 illustrates this cut generation approach. For the sake of easier illustration, commodities are represented by a single commodity number (rather than origin-destination nodes) and their origin and destination nodes are depicted explicitly in the figure.

In this example, we generate one cut by aggregating the individual cuts of the two commodities originated in region 1 and destined to region 5 (commodities 4 and 5); and one cut for the three commodities originated in region 8 and destined to region 3 (commodities 1, 2, and 3).

We define REG_{OD} as the set of all region pairs O and D that at least one commodity originates in O and destined to D, $r_{OD} \in REG_{OD}$. In our example, $REG_{OD} = \{(1,5),(8,3)\}$. Also, let C_{OD} be the set of commodities associated with region pairs O and D. Then, for our example, $C_{15} = \{4,5\}$ and $C_{83} = \{1,2,3\}$. These cuts are given as

$$\eta_{r_{OD}} \geq \sum_{[i,j]\in C_{OD}} \sum_{k\in\mathcal{R}} \sum_{l\in\mathcal{R}} (h_k q_{3_{ij}}^{kl} + h_l q_{4_{ij}}^{kl}) + \sum_{[i,j]\in C_{OD}} (q_{2_{ij}} + (1+\Omega) d_{ij} q_{5_{ij}}) \quad \forall\, r_{OD} \in REG_{OD}.$$

In our computational study, we show that the performance of this cut is impressive, and can manage both run time and memory usage very efficiently.

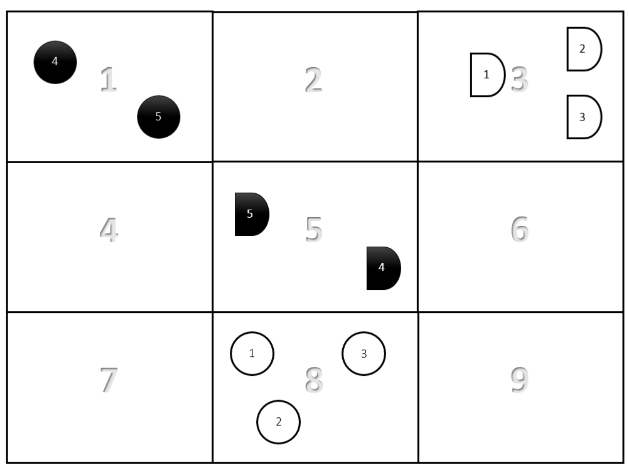

Figure 3.2: An illustration for ODReg cut disaggregation scheme

OReg Cut We use the structure of the previous cut type and define REG_O as the set of all regions that are the origins of at least one commodity, $r_O \in REG_O$, i.e., in our example, $REG_O = \{1, 8\}$. Then, the OReg cuts are

$$\eta_{r_O} \geq \sum_{i \in r_O} \sum_{\{j|[i,j] \in \mathcal{Q}\}} \sum_k \sum_l (h_k q_{3ij}^{kl} + h_l q_{4ij}^{kl}) + \sum_{i \in r_O} \sum_{\{j|[i,j] \in \mathcal{Q}\}} (q_{2ij} + (1+\Omega) d_{ij} q_{5ij}) \quad \forall\, r_O \in REG_O.$$

In general, adding multiple cuts can provide tighter bounds for the overall problem, however, too many cuts added to BMP can hamper its efficiency. The above ODReg and OReg cuts are motivated to provide less number, yet effective, Benders cuts but combining commodities which present similarities geographically in terms of the proximity of their origin and destination nodes. The effectiveness of these cut disaggregations is explicitly studied later in Section 3.3.2. It should be noted to implement the cut disaggregation scheme discussed the term η in Equation (3.27) is replaced by $\sum_{[i,j] \in \mathcal{Q}} \eta_{ij}$, $\sum_{i \in \mathcal{Q}_O} \eta_i$, $\sum_{r_{OD} \in REG_{OD}} \eta_{r_{OD}}$ and $\sum_{r_O \in REG_O} \eta_{r_O}$ for OD Cut, O Cut, ODReg Cut and OReg Cut, respectively.

3.2.6 Early Termination of BMP

In the conventional Benders decomposition implementation, the BMP is solved until reaching to optimality or a negligible gap between its upper and lower bound from in all iterations of the algorithm. Sometimes reaching to a negligible value of optimality for BMP is time-consuming due to diminishing improvements in optimality gap as branching progresses. In this study, for initial iterations of the algorithm, instead of solving the BMP until reaching to optimality, we terminate the BMP solving process upon reaching a predetermined higher level for optimality gap and consider the BMP's lower bound upon termination as the lower bound for MnP at that iteration. On the other hand, we use the last incumbent solution upon termination of BMP as input to the DBSP. After solving the BDSP, the summation of the objective function value of BDSP and the value of the integer part of BMP objective function (regarding the mentioned incumbent solution) provide an upper bound for the problem.

This technique increases the speed of the overall algorithm dramatically, although it may generate weaker lower bounds and Benders optimality cuts in the initial iterations of the algorithm. Specifically, we perform a stepwise BMP early termination optimality gap reduction. Based on our experiments, starting with 10% gap for the initial four iterations, then reducing to 4% for the next two iterations, and finally a 3% gap (which is acceptable gap% level for our overall algorithm) for the remaining iterations give the best results. This approach allows the initial cuts to be added quickly to reduce the set of feasible solutions to the master problem sooner.

3.2.7 Initial Bound Heuristics

A heuristic can be used to obtain a good initial upper bound as well as to add initial cuts to BMP to tighten the lower bound of the problem. For this purpose, we devise two heuristics built on the observation that the BPSP is the shortest path problem for each commodity when the RP locations are fixed.

SPH1 In the first heuristic, Shortest Path Heuristic 1 (SPH1) given in Algorithm 1, we solve the BPSP by using the Dijkstra's algorithm. To do so, we first assume that all the potential RP locations are open and find the shortest paths (in origin-RPs-destination sequence) for commodities' flows. We set the network such that the length of each arc is equal to its actual unit shipment cost which is determined based on the arc type (network or direct shipment arc type).

Once we find the cost-based shortest path for each commodity (which may be utilizing RPs or a direct shipment method), we check each shortest path distance to verify that the circuity constraint is not violated. If it is, then we remove the resulted solution and start the algorithm over to finally find the next shortest path solution until we find one cost-based shortest path whose corresponding distance does not violate the circuity constraint.

Finding shortest paths enables us to add one or a set of additional cuts to the BMP to strengthen the lower bounds and reduce its runtime. Again using the fact that the objective function of the BPSP is separable for each commodity $[i, j] \in \mathcal{Q}$ and it is equivalent to the cost-based shortest path problem for a fixed set of RPs, analogous to disaggregation schema for Benders cuts, we have the following **shortest path cuts** for our problem.

$$\eta_{ij} \geq SPCost_{ij} \qquad \forall\ [i,j]\ \in \mathcal{Q} \tag{3.48}$$

$$\eta_{i} \geq \sum_{\{j|[i,j]\in\mathcal{Q}\}} SPCost_{ij} \qquad \forall\ i \in \mathcal{Q}_O \tag{3.49}$$

$$\eta_{r_{OD}} \geq \sum_{[i,j]\in C_{OD}} SPCost_{ij} \qquad \forall\ r_{OD} \in REG_{OD} \tag{3.50}$$

$$\eta_{r_O} \geq \sum_{i\in r_O} \sum_{\{j|[i,j]\in\mathcal{Q}\}} SPCost_{ij} \qquad \forall\ r_O \in REG_O \tag{3.51}$$

where $SPCcost_{ij}$ represents the cost-based shortest path for commodity $[i,j] \in \mathcal{Q}$ when all RPs are available. Clearly, $SPCost_{ij}$ is the absolute least cost path for $[i,j]$ and thus, it is a lower bound on the optimal transportation cost for that commodity. Constraints (3.48), (3.49), (3.50), and (3.51) are formed similarly to the OD, O, ODRegt, and OReg Cuts in §3.2.5, respectively. These cuts are added to BMP in line 3 in SPH1.

In the next step, we create an array of size $|\mathcal{R}|$, namely "RP-UsageArray", which contains the number of times that each RP is used in the shortest paths obtained. Using this array, we first calculate an upper bound for the problem by adding the total activation cost of the RPs which are used at least once (having positive values in RP-UsageArray) to the total shortest paths' cost obtained. Afterward, we solve DBSP by fixing a group of RPs, having RP-UsageArray value greater than a predetermined threshold, namely "RP-UsageThreshold," as the open RPs (set $\mathcal{A}$, and add Benders optimality cuts to the initial BMP using the corresponding solution.

Algorithm 1 Shortest Path Heuristic 1 (SPH1)

1: Initialize set $\mathcal{A} = \emptyset$ and RP-UsageThreshold value
2: Solve BPSP using Dijkstra's algorithm with all RPs are available
3: Add shortest path cuts to the BMP
4: Create RP-UsageArray
5: Calculate UB
6: **for** $k \in \mathcal{R}$ **do**
7:     **if** RP-UsageArray$_k \geq$ RP-UsageThreshold **then**
8:         $\mathcal{A} = \mathcal{A} \cup RP_k$
9:     **end if**
10: **end for**
11: Solve DBSP considering the set $\mathcal{A}$ as the open RPs
12: Add Benders optimality cuts to BMP

SPH2 The second heuristic, called Shortest Path Heuristic 2 or SPH2, is given in Algorithm 2. We use this algorithm when the direct shipment is not allowed, i.e., TL shipments are allowed only on the RP network.

This algorithm is slightly different than Algorithm 1 in such a way that, after solving BPSP by Dijkstra's algorithm (finding the cost-based shortest paths) and updating the upper bound by considering the RPs used, we add cuts of the form $h_k = 1$ to BMP for those RPs that have shortest path usage value greater than the threshold RP-UsageThreshold. Then, we also add the surrogate constraints (§3.2.4) to the BMP and solve BMP to obtain the $\hat{h}.$ values which provides a collective set of open RPs. Moreover, shortest path cuts (3.48)-(3.51) are added to BMP as shown on line 3.

Algorithm 2 Shortest Path Heuristic 2 (SPH2)

1: Initialize RP-UsageThreshold value
2: Solve BPSP using Dijkstra's algorithm with all RPs available
3: Add shortest path cuts to the BMP
4: Create RP-UsageArray
5: Calculate UB and
6: **for** $k \in \mathcal{R}$ **do**
7:     **if** RP-UsageArray$_k \geq$ RP-UsageThreshold **then**
8:         Add cut $h_k = 1$ to the BMP
9:     **end if**
10: **end for**
11: Add surrogate constraints
12: Solve the BMP and update $\hat{h}.$ values
13: Remove the recently added cuts to the BMP at Step 8

Note that this solution to BMP does not provide a lower bound as it forces some RPs to be open. In later iterations, we remove the cuts added in line 8 from the BMP while we keep the surrogate constraints and the shortest path cuts. The attained $\hat{h}.$ values are used as the initial open RP locations to start Algorithm 3, line 3. It should be noted that circuity constraint consideration step in this algorithm can be done similar to that of SPH1.

3.2.8 Overall BD Implementation

Algorithm 3 presents the overall steps of the strengthened Benders decomposition. After initializing the algorithm's parameters and constants, the heuristics are utilized in line 2 to add initial cuts to BMP and tighten the bounds before starting the main algorithm's loop. In line 3, we establish the starting values for RP locations as dictated by the initial heuristic employed.

The main loop of the algorithm is started in line 4 for which two criteria, optimality gap and runtime - whichever is reached first, are employed to terminate the algorithm. By fixing all open RPs (or none, in the case of direct shipments allowed), we solve the DBSP in line

5 and update the current upper bound (UB) if necessary. The cut strengthening procedure, discussed in Section 3.2.3, is performed in line 6. Afterward, disaggregated Benders cut, presented in Section 3.2.5, using strengthened DBSP variable values are generated and added to BMP. The BMP, also including surrogate constraints (in case of no direct shipment)and the shortest path cuts added in heuristics along with Benders cuts, is solved to obtain a new set of $\hat{h}$ values and a new lower bound (LB). The iterations are continued in this fashion until a termination criterion is met. The terminating criteria are checked again in line 8 and if they are not met, the values of $\hat{h}$ are replaced by the recently achieved ones and we go to line 4. After termination of the algorithm, final routing decisions are made with the current open RPs in place by solving BPSP.

Algorithm 3 Strengthened BD Implementation

1: **initialize** $\epsilon = 0.03$, `optgap` $= 1.0$, `Runtime`$=0$, `Stoptime`$=7200$
2: Employ SPH1 or SPH2 if direct shipments are not allowed
3: If SPH1 is used in 2, set $\hat{h}_k = 0 \quad \forall k \in \mathcal{R}$;
   Otherwise, $\hat{h}_{\cdot}$ values are as provided by Algorithm 2 (SPH2)
4: **while** (`Runtime` $\leq$ `Stoptime` **and** `optgap` $> \epsilon$) **do**
5:     Solve DBSP and calculate UB and update if necessary
6:     Update the DBSP variables for strengthened Benders cuts
7:     Generate disaggregated strong Benders cuts and add it to BMP
8:     Solve BMP and update the LB
9:     Record `Runtime` and `optgap` $(= (1 - \text{LB}/\text{UB})$
10:     Update the $\hat{h}_{\cdot}$ by using the results of the recently solved BMP
11: **end while**
12: **return** $\hat{h}_k$, $\forall k \in \mathcal{R}$, solve BPSP and determine final routings.

3.3 Computational Study

In this section, we first describe our approach to generate experimental testbeds to be used to evaluate various aspects of our model and solution methodology. To this end, we assess the following three cases in our computational study:

With-direct-shipment-No-circuity DS-NC (Base case)

We consider this case as the base case of our experiments. The base case contains solving the MnP model without Constraint (3.6) which is the circuity constraint. To solve this model, we initialize Algorithm 3 by SPH1 (Algorithm 1) in its second step.

With-direct-shipment-With-Circuity-case DS-C

Considering the circuity constraint into the base case (DS-NC) yields DS-C case referring to the problem whose model is MnP. Solution procedure of this case is similar to DS-NC as given in Algorithm 3 with SPH1, but a circuity parameter Ω should be specified in line 1.

No-Direct-Shipment-No-Circuity-case NDS-NC

We consider this case to evaluate the effectiveness of our solution procedure when the direct shipment is not the option. We also disregard the circuity constraint, because our findings from two previous cases about including this constraint type are extendable over this case. To this end, if we remove the variable z from the MnP model in the base case that does not include circuity constraints, we obtain a new model in which the shipment is possible just through the network, i.e., no direct shipments allowed. In this situation, solution of BMP cannot guarantee the feasibility of BPSP, thus, the feasibility cuts are required. To reduce the negative effects of excessive feasibility cuts, we consider the new surrogate constraints proposed in Section 3.2.4 through using SPH2, Algorithm 2, in step 2 of Algorithm 3 to improve the lower bound attained and to facilitate generation the feasible solutions to BPSP.

After presenting experimental input data generation process, we first examine the effects of enhancements on the algorithmic performance using the first case above, and then, provide an analysis of results for varying input values for each case separately.

In the experiments presented in this section, we adopt the same stopping criteria, which include an optimality gap of 3% and a time limit of 7200 seconds, for our algorithm as well as the B&C implementation (as provided by CPLEX). We solve five test instances for each class described below and report the average values over these instances for the cases. In the tables below, the "Gap%" column reports the final gap percentage achieved and it is calculated as $[(UB - LB)/UB] * 100$ and the column "T(s)" refers the runtime in seconds and, since the algorithms check the gap discretely, it may happen that the time stopping criterion is met in a slightly larger value than 7200 seconds.

To solve all the presented models, we use CPLEX 12.6.1, and all of the experiments are conducted using C++ on machines with an Intel Core i7-4790 CPU at 3.6 GHz, 32 GB RAM running 64-bit OS.

3.3.1 Data Generation

We generate our random test problem such that sensitivity analysis can be done on the different parameters of the problem. As a geographical region , we consider a rectangular region with dimension 150×100 (width $\times$ height). To represent TL origin and destination nodes, we randomly generate $|\mathcal{N}|$ clustered point coordinates and, to represent potential RP locations, $|\mathcal{R}|$ uniformly distributed point coordinates in the same region.

To have the clustered point locations, where each cluster can represent an area of dense population such as a city or metropolitan area, we divide the whole area by 25 sub-rectangles (each is 20×25) and generate 75% of the origin and destination nodes in five of them picked randomly and the remaining 25% uniformly all over the main area to keep diversity between points. Potential RP locations are also generated uniformly in the main area. We consider $|\mathcal{N}|$ values of 160, 180, 200 and 220, and $|\mathcal{R}|$ values of 25, 30, and 35. Therefore, eight distinct problem classes, Class 1 (C1) to Class 8 (C8), are specified by $(\mathcal{N}, \mathcal{R})$ as (160, 25), (160, 30), (180, 25), (100, 30), (200, 30), (200, 35), (220, 30), (220, 35), respectively.

We assume D percent of the pairs of the commodity nodes requires a commodity flow. The basic value for this density measure is 30%, but we also consider 35% for further analyses. Then, the total number of commodities $|\mathcal{Q}|$ is given by $|\mathcal{N}|(|\mathcal{N}|-1)D$.

To have more realistic test problems, we categorize the commodities into three groups based on the direct shipment distance between their origin and destinations as long, medium, and short range commodities. Based on the geographical region scales considered, the distance values 30 and 60 are employed as the threshold levels between short-to-medium and medium-to-long distance commodities, respectively. To obtain these three sets, the node pairs are first sorted in descending order of their Euclidean distance and the list is equally divided into three groups using the threshold values. Letting the triplet (L, M, S) represent the share of commodities selected from the long-, medium-, and short-range commodity groups on a percentage basis, we then generate the commodity set $\mathcal{Q}$ by randomly selecting $L\%|\mathcal{Q}|$, $M\%|\mathcal{Q}|$, and $S\%|\mathcal{Q}|$ different commodities from the long, medium, and short range groups, respectively. In our numerical study, we mainly consider (L, M, S) values of (80, 10, 10), but also examine (60, 20, 20) to do a sensitivity analysis.

The demand w_{ij} (in TLs) for each commodity is randomly determined using uniform distribution U[10, 20]. In terms of the cost parameters, the fixed cost of locating an RP is assumed to be 150,000 for C1 and C2, 225,000 for C3 and C4, 300,000 for C5 and C6 and 375,000 for C7 and C8; i.e., the fixed cost is increased with increasing number of commodities ($|\mathcal{N}|$) served to reflect the fact that more expensive facilities are required to be able to serve a higher number of TLs. Furthermore, all of the instances utilize the values 1, 2 and 3 for unit transportation costs t_1, t_2, and t_3, respectively. Finally, for local and lane tour length distance constraints, we consider (Δ_1, Δ_2) values of (20, 40) mainly and (25, 50) for further analysis.

Table 3.2 summarizes our test problem classes and their characteristics. We note that the location of the nodes in classes C1, C3, C5, C7 are kept the same in C2, C4, C6, C8, instance-by-instance, respectively, while the required number of potential RP locations are added to the problem randomly to increase the size as needed for the latter classes.

Table 3.2: The test instances classes ($(L, M, S) = (80, 10, 10)$; $(\Delta_1,\ \Delta_2) = (20, 40)$; $D\% = 30$)

Class	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{R}\rvert$	f	$\lvert\mathcal{Q}\rvert$
C1	160	25	150000	7632
C2	160	30	150000	7632
C3	180	25	225000	9666
C4	180	30	225000	9666
C5	200	30	300000	11940
C6	200	35	300000	11940
C7	220	30	375000	14454
C8	220	35	375000	14454

3.3.2 Numerical Results on Algorithmic Performance

In this subsection, we present computational results to examine the enhancement components of our algorithm for improved performance.

- First, to show the effectiveness of the strengthening Benders cuts as described in Section 3.2.3, we run the BD-ODReg (BD in Algorithm 3 with OD-Reg cuts) with and without strengthened cuts and compare the runtimes and the number of optimality cuts required utilizing four larger classes of test instances. Table 3.3 reports the results of this experiment. The left part of the table reports the results of using strengthened cuts instead of regular ones in the BD-ODReg algorithm and right part shows the results of the having regular Benders cuts. In addition to runtimes, we also report the average number of optimality cuts required, "No. of opt. cuts req.," to solve the instances.

  As it can be seen, adding strengthened Benders cuts can decrease the numbers of optimality cuts required at least by one cut on average. In terms of runtimes, the algorithm with strengthened cuts can perform faster by about 11% on average. The effectiveness of including the strengthened cuts can be more emphasized by mentioning that by adding them instead regular ones, 45% of the test instances of the Table 3.3 can be solved faster, while only 15% of them are being solved faster with regular Benders cuts, and for 40% both have the same performance. Based on these observations, we recommend using strengthened cuts and use them in all experiments in this study.

Table 3.3: Effectiveness of Strengthened Benders Cuts

Class	Str. Benders Cuts		Reg. Benders Cuts	
	T(s)	No. of opt. cuts req.	T(s)	No. of opt. cuts req.
C5	619	10	722	11
C6	1424	15	1506	16
C7	1241	11	1384	12
C8	3598	19	4157	20

- Second, to demonstrate the effectiveness of SPH1, we conduct an experiment using classes C5 and C6 solved by using BD-ODReg algorithm with and without the this heuristic employed in its initial stage. Table 3.4 shows the results of this experiment.

  SPH1 seems very effective since it can decrease the runtime by 42% on average. The parameter RP-UsageThreshold is set to 0.5 since this value provided the most desirable results (this may require testing different values for other instances). Another interesting result of using SPH1 is that after implementing this heuristic, the initial gap of the main BD algorithm was less than 20% for all test instances and, for most of them, it was less than 10%.

Table 3.4: Effectiveness of SPH1

Class	BD-ODReg (with SPH1)		BD-ODReg (w/o SPH1)	
	Gap%	T(s)	Gap%	T(s)
C5	2.72	619	2.66	1105
C6	2.56	1424	2.78	2339

- Third, to evaluate the performance of cut disaggregation schema presented in Section 3.2.5, we solve the instances of four test classes using the BD algorithm by naming it as BD-ODReg, BD-OD, BD-OReg, and BD-O to refer to the cut type employed. Table 3.5 summarizes the results of this experiment.

  BD-ODReg clearly has the best performance in runtimes and the optimality gaps in general. Among the other three algorithms, BD-OD has the best performance while its runtime is about 150% worse than BD-ODReg over all test classes. This ratio for BD-OReg and BD-O are 490% and 420%, respectively.

  In terms of the number of adding cuts in each iteration, this amount for BD-ODReg is about 38% of that of BD-OD for C3 and C4 and 33% for C5 and C6. Also, by increasing the size of the problem from C4 to C6 (C3 to C5) the number of adding cuts grow by 24% for BD-OD, whereas this value for BD-ODReg is only 8%.

Based on these two important points mentioned above, we can say that the memory management of BD-ODReg is significantly better, especially for large-sized instances. Although BD-Oreg and BD-O algorithms have higher runtimes than BD-ODReg, they can improve the performance over conventional BD dramatically with relatively small number of extra Benders cuts added in each iteration. In terms of runtime, BD-O is faster than BD-OReg by about 13% on average, but it is adding 123% more cuts than BD-OReg on average in each iteration.

Table 3.5: Comparing different Benders cuts performance

Class	BD-ODReg			BD-OD			BD-Oreg			BD-O		
	No. of Cuts	Gap%	T(s)	No. of Cuts	Gap%	T(s)	No. of Cuts	Gap%	T(s)	No. of Cuts	Gap%	T(s)
C3	3672	2.48	236	9666	2.48	377	78.4	2.80	1193	168.2	2.76	1021
C4	3672	2.54	500	9666	2.34	1671	78.4	3.58	4588	168.2	2.98	3563
C5	3959	2.72	619	11940	2.18	1685	77.2	3.22	4765	178.6	2.82	3825
C6	3959	2.56	1424	11940	2.54	3138	77.2	4.72	5908	178.6	4.12	5915

- Finally, to find a reasonable number of partitions that we can use to generate cuts in BD-ODReg, we test various number of partitions using the classes C5 and C6. For this purpose, we divide each side of the geographical area by 10, 20, and 30 resulting in 100 10×10 regions, 400 20×20 regions, and 900 30×30 regions, respectively. We compare the performance of BD-ODReg using these different number of regions in Table 3.6.

  We observe that the 20×20 case gives the best results in terms of runtime. The 10×10 case, although gives a higher runtime by 9%, the number of cuts added in each iteration is significantly less than that of 20×20 and 30×30 cases. In the rest of this study, we adopt the 20×20 case as the default case, but one can use 10×10 or another less granular setting to manage memory more efficiently. The 30×30 case runtime is higher than two other cases due to BMP solution time since it adds a higher number of cuts each iteration.

Table 3.6: Evaluating the effect of the number of regions on the performance of BD-ODReg

Class	100 10×10 regions			400 20×20 regions			900 30×30 regions		
	Gap%	T(s)	No. of Cuts	Gap%	T(s)	No. of Cuts	Gap%	T(s)	No. of Cuts
C5	2.64	673	517	2.72	619	3959	2.46	1025	6927
C6	2.76	1561	517	2.56	1424	3959	2.42	2377	6927

3.3.3 Analysis on Input Parameters

In this subsection, we summarize analysis results for the three cases outlined in the beginning of this section.

In the tables presented in this section, we use the following notation to report the results. The number of RPs used in the final solution is presented in the "No. of open RPs" column. The average fixed cost percentage of total cost for activation the RPs are given in the "Fixed cost%" column. The "Drct. ship. cost %" and "Drct. ship. %" present direct shipment cost percentage of total cost and the number of directly shipped commodities as a percentage of total shipments, respectively. The "Ave. Cir. %" column shows average circuity over all TL shipments in the final solution.

3.3.3.1 Regular Case - Direct Shipment without Circuity

In this base case, we consider the RP network design with direct shipments without taking the circuity control effects into account. Table 3.7 shows the results of the BD algorithm when we use ODReg cut type and B&C implemented by Cplex for different test problem. As before, five test instances of each class are solved and the average results are presented for each measure.

The BD-ODReg algorithm can efficiently solve all the instances under the stopping criteria employed, while the B&C cannot achieve this for one of the test instances in C5, four in C6, three of Class 7 and four of Class 8. There is a significant difference between runtime of BD-ODReg and B&C in all test classes. Although the average gap in classes with smaller size instances is smaller with B&C, this changes quickly and significantly for large size instances. For the instances that can be solved using both approaches, the average runtime for BD-ODReg is about 15% of that of B&C, and none of these test instances can be solved faster than BD-ODReg using B&C.

We observe that the number of RPs used in the final solution does not increase by increasing number of commodities. We can conclude that for a geographical region as that one we consider, having so many RP possible locations are not necessarily required, and we can easily manage all the demands using a relatively small number of open RPs. The average fixed cost percentage of the total cost is about 14% of total cost. This value is reasonable given that the RP locations are not expected to provide services such as load splitting, sorting, loading, etc.

The other two important measures reported are the average of direct shipment cost percentage of total cost, and the average of the number of commodities demands satisfied using direct shipment. The first one has a value of 5.5% over all classes, and the second one is 6.4%. Average circuity, for those commodities that use network shipment, is around 20% for all the test classes. This value shows that to implement a relay network setting to satisfy all the demands, we only need to travel 20% more than the sum of all point to point distances. This additional distance seems reasonable and expected in terms of transportation companies point of view.

Table 3.8 presents more detailed routing information regarding four classes of test instances, C3, C4, C5 and C6. The first section of this table namely "Ave. No. of RP used%" reports the average percentage of the number of RPs visited by shipments for those commodities using the network to be shipped. We observe that about 40% of the commodities utilize 3 RPs on their route to the destination. In other words, for the majority of shipments we only need to utilize four drivers, one per leg. Also, 26% of the commodities use 4 RPs and 8%, 14% and 10% of them use 1, 2 and 5 RPs, respectively, on average. Only less than 3% of all commodities visit more than 5 RPs while they are shipped to their destination.

Table 3.7: Results of BD-ODreg and B&C for different classes of test problems

Class	B&C		BD-ODReg						
	Gap%	T(s)	Gap%	T(s)	No. of open RPs	Fixed cost %	Drct. ship. cost%	Drct. ship.%	Ave. Cir.%
C1	0.96	510	2.60	142	14	11.6	7.2	8.0	24.2
C2	1.28	1398	2.46	235	14	12.0	3.8	5.2	22.0
C3	0.30	1088	2.48	236	14	14.6	4.8	5.0	17.8
C4	0.26	3471	2.54	500	14	15.0	2.4	3.6	17.4
C5	2.49	5075	2.72	619	12	13.4	7.2	8.2	22.8
C6	9.81	6634	2.56	1424	12	13.2	3.0	3.8	19.0
C7	5.38	6686	2.72	1241	14	15.8	6.8	8.8	21.0
C8	13.20	7998	2.50	3598	13.2	15.8	8.4	8.6	19.2

Table 3.8: Detailed routing information

Class	Ave. No. of RP Used %						Rat. Drct. Ship.%			Drct. Ship. %		
	1	2	3	4	5	>5	L	M	S	L	M	S
C3	8.2	15.2	42.4	24.0	9.4	0.8	2.8	5.4	22.6	37.6	10.4	52.0
C4	8.2	16.0	42.8	23.2	9.0	0.2	1.2	3.4	23.2	19.2	9.6	71.2
C5	8.0	12.4	30.2	29.0	13.4	7.0	5.2	15.0	25.0	40.4	21.6	38.0
C6	8.0	12.4	39.6	27.2	9.8	2.6	1.6	8.2	20.8	22.2	23.2	54.6

The column labelled as “Rat. Drct. Ship.%” reports the average rational percentage of each type of commodity based on distances shipped using direct shipment method. On average, only less than 3% of long-distance commodities are shipped directly, whereas about 23% of short distance and 8% of medium distance commodities are shipped in this way. We can conclude that direct shipment is a better choice for short distance commodities and embedding this option to the model gives this flexibility to the model to decide not to use network shipment whenever it is not beneficial. We can explain this result by the relatively low cost of direct shipment for short distance, as opposed to following a longer route via the RPs. The column "Drct. Ship.%" shows the average percentage of different types of commodities (L, M, and S) in total directly shipped ones. On average over test classes, of all the direct shipments, about 30%, 16% and 54% are from long, medium and short distance commodities, respectively. This confirms that the interest of the system in transporting short distance shipments directly is higher than that of long-distance ones.

In order to perform a sensitivity analysis on our BD algorithm, we change the parameters of the test problems, one input parameter at a time, and solve them again. These changes are done in three directions. In the S1 case, we increase the (Δ_1, Δ_2) values to $(25, 50)$. Increasing these parameters will increase the size of the problem dramatically with the increasing number of possible routes to deliver the demands. The second case, S2, is about increasing the density D to 35%and thus increasing the number of commodities. Having a higher number of commodities to be delivered will also increase the size of the instances. Finally, in S3, we change the values for triplet (L, M, S) as $(60, 20, 20)$, while maintaining all the nodes locations of the instances the same as the original ones. This change would vary the nature of the commodities and optimal routes and consequently change RP locations. Table 3.9 presents the results of this analyses.

In S1, the runtime increases for C5 by 19% and C6 by 113% when we use BD-ODReg to solve the problems. All the test instances of these two classes are solved and the average gap is similar to the DS-NC case. The S1 case influences B&C more; only one of the test instances is solvable in C5 and none of the instances in C6 is solvable using B&C. The main reason for this is that increasing the Δ values increases the potential routes through the RPs dramatically. That's why the effect of this change is more pronounced for C6.

Similarly, in S2 case, increasing density increases the size of the problem and consequently, the runtimes. The runtime of BD-ODReg increases by 18% and 61% for C5 and C6, respectively. All the test instances in these two class can be efficiently solved by BD-ODReg, but one of them in C5 and two in C6 cannot be solved using B&C.

The results for S3 are interesting because the runtime decreases for C5 by 22% and increases for C6 by 62%. Generally, we can say that changing the grouping of commodities can change the runtime specially when we increase the number of potential RP locations. We can see the same behavior when we use B&C to solve these problems. All the test problems can be solved by BD-ODReg, but two of C5 and four of C6 are not solvable by B&C.

Table 3.9: Results of sensitivity analysis for the DS-NC case

Class	S1: (Δ_1, Δ_2)=(25,50)				S2: D=35%				S3: (L,M,S)=(60,20,20)			
	BD-ODReg		B&C		BD-ODReg		B&C		BD-ODReg		B&C	
	Gap	T(s)	Gap%	T(s)	Gap	T(s)	Gap%	T(s)	Gap	T(s)	Gap%	T(s)
C5	2.26	735	16.58	6724	2.66	734	1.9	5025	2.38	485	2.54	4774
C6	2.56	3036	27.38	7821	2.54	2294	7.44	6711	2.58	2309	13.86	6658

3.3.3.2 *Direct Shipments with Circuity Case*

In this subsection, we examine the performance of our BD-ODReg algorithm by also considering the circuity constraint (3.6). We vary the circuity parameter Ω to investigate its effect on the runtimes and optimal solution characteristics. Specifically, we tighten the associated constraint by changing Ω values as 1, 0.5 and 0.25 with results reported in Table 3.10 for two classes, C5 and C6, of test problems solved with BD-ODReg and B&C.

Table 3.10: Results of solving the model considering the circuity constraint

Class	Ω	B&C		BD-ODReg						
		Gap%	T(s)	Gap%	T(s)	No. of open RPs	Fixed cost %	Drct. ship. cost %	Drct. ship. %	Ave. Cir.%
C5	1.00	4.00	5284	2.46	498	11.2	11.8	7.6	9.2	21.4
C6		13.04	6784	2.58	1336	12.0	13.2	3.4	5.6	18.4
C5	0.50	6.88	5213	2.64	543	10.8	11.2	15.0	16.6	18.8
C6		9.94	6735	2.10	1445	11.4	12.4	6.6	8.8	15.6
C5	0.25	3.45	6262	2.36	1052	12.2	12.2	40.8	36.6	11.6
C6		12.75	7243	2.88	2902	11.2	11.8	33.2	30.0	10.8

Tightening the Ω does not have a significant effect on the runtime of the BD-ODReg when Ω is changed from 1 or 0.5 for our problem instances in general. However, once Ω is set to 0.25, we see a notable effect in runtime which increases by 87% on average. Other impacts of this change include changes in direct shipment percentage and direct shipment cost percentage that are significantly higher than the $\Omega = 1$ case due to more TLS being transferred as direct shipment under limiting circuity allowance.

This can be explained by observing that, in Table 3.7, the average circuity for all the test instances in C5 and C6 is about 20% while it contains commodities that may follow significantly circuitous routes to avoid costly direct shipments. Setting the Ω level (worst-case circuity level) to a level close or less than the basic average circuity changes routings of TLs in a way that more commodities are shipped directly which, in turn, lead to decreased number of RPs used and lower average circuity for network shipped commodities. In terms of the runtimes, B&C runtime is affected more negatively than BD-ODReg by the reduced changing the Ω level, especially for C6.

To further investigate the impact of Ω on the algorithmic performance under varying input parameters, we present a sensitivity analysis on a number of parameters similar to what we did in the DS-NC case with parameter groups S1, S2, and S3 as reported in Table 3.9. The results presented in Table 3.11 definitely align with the interpretations previously discussed in Table 3.9.

Table 3.11: The results of sensitivity analysis on the model with circuity constraint (DS-C case)

Class	Ω	S1: (Δ_1, Δ_2)=(25,50)				S2: D=35%				S3: (L,M,S)=(60,20,20)			
		BD-ODReg		B&C		BD-ODReg		B&C		BD-ODReg		B&C	
		Gap%	T(s)	Gap%	T(s)	Gap%	T(s)	Gap%	T(s)	Gap%	T(s)	Gap%	T(s)
C5	1.00	2.52	1026	21.24	6773	2.45	611	3.06	5007	2.10	577	2.56	4883
C6		2.66	2308	27.34	7972	2.46	1682	8.60	6932	2.38	1795	13.84	6790
C5	0.50	2.50	1226	20.60	7088	2.30	611	6.78	5383	2.66	616	6.56	5378
C6		2.64	2801	27.26	7783	2.50	1906	9.58	7660	2.68	2329	11.50	7292
C5	0.25	2.52	2347	19.86	6993	2.32	759	4.04	5385	2.32	1135	6.32	6327
C6		3.00	5091	25.58	7670	2.80	3413	9.12	7644	2.84	3419	13.94	7586

3.3.3.3 No Direct Shipment, No Circuity Case

To evaluate the performance of the proposed BD-ODReg, when the direct shipment is not allowed, we design a different experiment. To do so, we implement the proposed BD-RDReg on four classes of test problems. In Step 2 of Algorithm 3, we utilize SPH2. Having surrogate constraint may increase the memory usage, thus we pick four smallest classes of test instances. Table 3.12 demonstrate the results of our implementation when we do not have direct shipment possibility.

Table 3.12: Results of solving the model without direct shipment

Class	B&C		BD-ODReg					BD-ODReg (without SPH2)		
	Gap%	T(s)	Gap%	T(s)	No. of Feas Cuts	No. of open RPs	Fixed cost %	Gap%	T(s)	No. of Feas Cuts
C1	0.94	907	2.64	160	0.2	11.6	11.2	2.68	313	2.0
C2	0.88	1303	2.40	393	0.8	14.0	12.0	2.06	644	2.6
C3	1.33	1350	2.72	359	0.2	13.6	14.4	2.11	483	1.4
C4	1.50	2089	2.60	641	0.2	13.6	14.4	2.33	1720	1.8

First, we compare the results of BD-ODReg and B&C implemented by Cplex. Considering the stopping criteria, BD-ODReg can solve all the tested instances in an interestingly shorter time. The runtime of BD-ODReg is about 27% of the runtime of B&C. In the column "No. of Feas. Cuts" we show the average of feasibility cuts used to solve the problem. As can be seen, on average we only need 0.35 feasibility cuts to solve a test instance implying we do not need so many feasibility cuts to solve such problems. This happens because of the surrogate constraint used. We can conclude that the proposed surrogate constraints, not only are easy to use but also they can almost guarantee the feasibility of BPSP. The runtime of the algorithm without surrogates is high, because of having so many feasibility cuts, so we do not report it. The number of open RPs and fixed cost percentage in the final solution are almost similar to what we reported in Table 3.7. Also, to evaluate the performance of SPH2 we compare the result of BD-ODReg with and without this heuristic (but with the surrogate constraints). As it is obvious, SPH2 performs well in terms of reducing the runtime and decreasing the number of required feasibility cuts to solve the instances. On average, the runtime of the algorithm with SPH2 is about 49% of its runtime without this heuristic. Also, the number of needed feasibility cuts decrease from 1.95 to 0.35 when we use SPH2, which is helpful to solve the instances in a shorter time.

3.4 Conclusion

In this study, we suggest utilizing the relay points through designing a well-structured relay network to alleviate the high driver turn over rate issue in the truckload transportation industry properly. To do so, we consider the design of a relay network having two new assumptions, parallel shipment and multiple assignments generalizing the problem and decreasing the overall cost. Giving the chance of using direct shipment method to a number of commodities, which shipping them through the network impose an extremely higher cost to the system, would decrease the final cost of transportation. Also, considering direct and network shipment options, concurrently, increase the flexibility of the system in terms of meeting the promise dates and managing the overall traveled distances. On the other hand, allocating the non-RPs to RPs on a commodity-based system can decrease the total cost by optimizing the total traveling distances, and results in a well-structured network.

We propose a mixed-integer programming formulation for the problem and devise an efficient solution approach based on Benders decomposition. Because of inefficiencies in the conventional Benders' decomposition to solve the proposed model, we employ several algorithmic improvements. In this regard, we propose an effective cut disaggregation scheme (ODReg), incorporate strengthened Benders' cuts, suggest a number of surrogate constraints benefiting the integrality property and stepwise early termination of Benders master problem, and devise a shortest path based heuristic to embed a number of initial cuts and tighten starting bounds. Our computational studies show that the performance of the proposed approach is very good for different sizes of instances such that on average its runtime is less than that of B&C, implemented by Cplex, by about 85% for those test instances which are solvable by both Cplex and proposed algorithm. It should be noted that more than 70% of large size instances are not solvable by Cplex. Also, our algorithm's memory management is effective such that it can manage the memory to solve the large size instances properly.

We assess the problem and the algorithm in three different cases, the *DS-NC case* depicting the situation that we disregard the circuity constraint, the *DS-C* case considering the problem with circuity constraint, and *NDS-NC* case representing the problem without direct shipment option to ship the commodities. We also examine various problem and algorithm parameters on the algorithmic performance and solution characteristics for different cases. The inputs including local and lane tour lengths, distance distribution of the commodities by differentiating the mix of long-medium-short coefficients, the density of the positive demands, also for "*with circuity*" case, the circuity allowable level.

In the context of network characteristics, we show that we can easily satisfy all the demands using a relatively small number of open RPs. The other two important measures reported are the average of direct shipment cost percentage of total cost and the average of the number of commodities demands satisfied using direct shipment. The average of the first index for all classes is 5.5% and for the second one is 6.4%. These two values demonstrate that although the direct shipment method can be used to facilitate the final routes of satisfying the demands, the model tries to meet the demands through the network shipment as much as possible to have a lower final cost. Using the direct shipment method causes increasing the total cost of the system implicitly because of its effect on increasing the driver turnover rate (we quantify and embed it to the per-unit direct shipment cost parameter). In this regard, we show that the majority of the commodities shipped using direct shipment approach are a type of short-distance commodities. Average circuity, for those commodities that use network shipment, is around 20% for all the test classes. This value shows that to implement a relay network setting to satisfy all the demands, on average, we only need to travel 20% more than the sum of all point to point distances. In this respect, we show that the majority of these commodities only visit three RPs to reach to their destinations. In other words, for the majority of shipments, we only need to utilize four different drivers. It should be noted that a relatively small group of commodities visit more than five RPs on their way to the destination.

Chapter 4

A Progressive Hedging-Strengthened L-Shaped Method for Relay Network Design Problem Under Uncertainty (P2)

In the truckload transportation industry, having uncertainty in the network's demands is inevitable. To address this situation, we should investigate designing the RP-network under uncertainty in commodities' demand. We use the two-stage stochastic programming approach to model the RP-network designing problem in this situation. The setting of our problem of interest builds on the deterministic RP-network design problem addressed in the previous chapter. In this chapter, we extend this model by considering uncertainty in commodities' demands. We build a two-stage stochastic programming model in which the first stage corresponds to design decisions which are optimal locations of the RPs and the second stage addresses finding the optimal solutions for a set of commodity routing problems each corresponding to a scenario generated to capture the uncertainty in demand. In other words, in our two-stage stochastic program, we address design decisions, which should be made here and now, in the first stage, and the decisions after realization of uncertainties, in the second stage. Also, we suggest an exact solution approach to solve the problem efficiently, specially for large size instances.

4.1 Problem Definition and the Model

The underlying setting of the problem follows from the one introduced in the previous chapter. Thus, below we discuss uncertainty representation for out problem in more detail building on the existing setting.

4.1.1 Uncertainty Representation

The decisions about the locations of the RPs are typically made at a point in time when the commodities' demands are not known with certainty. We consider this context in the problem formulation presented in Section 4.1.2 by letting each demand to be a random variable. On the other hand, the routing and assigning decisions can be made after realizing the actual demands. To have the described environment called two-stage stochastic programming (2-SP) we consider the following assumptions about the random variable $\tilde{\omega}$, representing the random demand.

ASSUMPTION 1. The random variable $\tilde{\omega}$ has a discrete distribution (with a finite support), say $\{\omega^1, ..., \omega^S\}$, with probabilities $Pr(\tilde{\omega} = \omega^1) = p^1$, ... , $Pr(\tilde{\omega} = \omega^S) = p^S$.

There is a realization of random demand $w(\omega^s)$ corresponding to each scenario ω^s. To have this, we consider a set of random demand scenarios, each of which represents a special behaviour of the system. For the sake of simplicity, we show $w(\omega^s)$ by w^s.

4.1.2 Mathematical Model

Before developing a 2-SP for the problem, we first introduce the notation and the decision variables.

Sets:

$\mathcal{N}$	set of commodity origin/destination nodes, $i,\ j \in \mathcal{N}$
$\mathcal{R}$	set of potential RP nodes, $k,\ l \in \mathcal{R}$
Q	set of commodities, $[i,j] \in Q$
$\mathcal{S}$	set of scenarios, $s \in \mathcal{S}$

Parameters:

$\tilde{w}_{ij}$	random variable for total demand for commodity $[i,j]$
w^s_{ij}	total demand for commodity $[i,j]$ for scenario s, a realization of $\tilde{w}_{ij}$
p^s	probability of occurrence of scenario s
d_{kl}	distance between node k and node l
t_1	transportation cost between RPs and non-RP nodes per-unit demand per-unit distance
t_2	transportation cost between two RPs per-unit demand per-unit distance
t_3	transportation cost between two non-RPs per-unit demand per-unit distance
f_k	fixed cost of locating an RP at node $k \in \mathcal{N}$

Δ_1	allowable distance between a non-RP node and an RP
Δ_2	allowable distance between two RP nodes

Decision Variables:

x_{ij}^{kls}	1 if node i is assigned to the RP at node k and node j is assigned to the RP at node l to ship commodity $[i,j] \in Q$ using the RP network, for scenario s; $i,j \in \mathcal{N}$
y_{ij}^{kls}	1 if commodity $[i,j] \in Q$ uses the arc (k,l); $k,l \in \mathcal{R}$, for scenario s; $i,j \in \mathcal{N}$
z_{ij}^{s}	1 if commodity $[i,j] \in Q$ shipped directly from node i to node j, for scenario s; $i,j \in \mathcal{N}$
h_k	1 if RP k, $k \in \mathcal{R}$ is used, 0 otherwise

In our 2-SP, the first-stage or here-and-now decisions, which to be made before realizing the uncertainty, address the design characteristics of network and the second-stage or wait-and-see decisions, which to be made after resolving the uncertainty through realization of a certain scenario, determine the origins/destinations-to-RPs assignments and routing information. The 2-SP aims to find a first stage solution which has the best performance over average of all scenarios. This is achieved by minimizing the total cost containing the RPs fixed cost and expected cost of transportation. First, we show the overall model in a deterministic equivalent form as follows.

[DEF]

$$\begin{aligned}\text{Min} \quad & \sum_k f_k h_k + \sum_s p^s \Big(\sum_i \sum_j \sum_k \sum_l t_1 (d_{ik} + d_{jl}) w_{ij}^s x_{ij}^{kls} + \sum_i \sum_j \sum_k \sum_l t_2 d_{kl} w_{ij}^s y_{ij}^{kls} \\ & + \sum_i \sum_j t_3 d_{ij} w_{ij}^s z_{ij}^s \Big) \end{aligned} \tag{4.1}$$

subject to

$$\sum_{\substack{m\in\mathcal{R}\\m\neq k}} y_{ij}^{mks} + \sum_{\substack{m\in\mathcal{R}\\m\neq k}} x_{ij}^{kms} = \sum_{\substack{m\in\mathcal{R}\\m\neq k}} y_{ij}^{kms} + \sum_{\substack{m\in\mathcal{R}\\m\neq k}} x_{ij}^{mks} \quad \forall\, [i,j] \in Q,\ k \in \mathcal{R}, s \in \mathcal{S} \tag{4.2}$$

$$z_{ij}^{s} + \sum_{k\in\mathcal{R}} \sum_{l\in\mathcal{R}} x_{ij}^{kls} = 1 \quad \forall\, [i,j] \in Q, s \in \mathcal{S} \tag{4.3}$$

$$x_{ij}^{kls} \leq h_k \quad \forall\, [i,j] \in Q,\ \forall\, k,\ l \in \mathcal{R}, s \in \mathcal{S} \tag{4.4}$$

$$y_{ij}^{kls} \leq h_l \quad \forall [i,j] \in Q,\ \forall\, k, l \in \mathcal{R},\ k \neq l, s \in \mathcal{S} \tag{4.5}$$

$$x_{ij}^{kls} \geq 0 \quad \forall\, [i,j] \in Q,\ \forall\, k,\ l \in \mathcal{R}, s \in \mathcal{S} \tag{4.6}$$

$$y_{ij}^{kls} \geq 0 \quad \forall\, [i,j] \in Q, \forall\, k,\ l \in \mathcal{R},\ k \neq l, s \in \mathcal{S} \tag{4.7}$$

$$z_{ij}^{s} \geq 0 \quad \forall\, [i,j] \in Q, s \in \mathcal{S} \tag{4.8}$$

$$h_k \in \{0, 1\} \quad \forall\, k \in \mathcal{R} \tag{4.9}$$

The first term of the objective function measures the total fixed costs associated with locating the RPs. The second part represents the expected value of all transportation costs of all the scenarios in which the first part is for transportation between origin/destination nodes and RPs, the second term for between RPs transportation and the last one for direct shipment costs. Constraints (4.2) are the flow conservation constraints for each node, commodity and scenario. Constraints (4.3) guarantee that the TLs are sent through the relay network or direct shipment to satisfy the demands. Constraint sets (4.4) and (4.5) are the linking constraints and ensure activating the required RPs. Constraints (4.6) to (4.8) originally enforce the *x, y,* and *z* to be binary. However, as we discussed in Section 3.1.1, the model has the integrality property since we do not consider any capacity limitation on the links of the network, hence these variables will be integral with values 0 or 1, and the model can be solved as a mixed-integer program (MIP). Constraints (4.9) consider the binary requirement of the variable *h*. Also, we present the proposed model in a two-stage stochastic program separated form, which we call SMnP, as follows.

$$\textbf{[SMnP]} \qquad \text{Min} \quad \sum_{k}(f_k h_k) + \mathbf{E}[\mathcal{Q}(h, \tilde{w})] \tag{4.10}$$

subject to

$$h_k \in \{0, 1\} \quad \forall\, k \in \mathcal{R}. \tag{4.11}$$

where recourse $\mathcal{Q}(h, w^s)$ for a particular realization $s \in \mathcal{S}$ is the optimal solution of the following linear program:

$$\begin{aligned}\mathcal{Q}(h, w^s) = \ \textbf{Min} \sum_{[i,j]\in Q}\sum_{k}\sum_{l} t_1(d_{ik} + d_{jl})w_{ij}^s x_{ij}^{kls} + \sum_{[i,j]\in Q}\sum_{k}\sum_{l} t_2 d_{kl} w_{ij}^s y_{ij}^{kls} \\ + \sum_{[i,j]\in Q} t_3 d_{ij} w_{ij}^s z_{ij}^s \end{aligned} \tag{4.12}$$

subject to

$$\sum_{\substack{m\in\mathcal{R}\\ m\neq k}} y_{ij}^{mks} + \sum_{\substack{m\in\mathcal{R}\\ m\neq k}} x_{ij}^{kms} = \sum_{\substack{m\in\mathcal{R}\\ m\neq k}} y_{ij}^{kms} + \sum_{\substack{m\in\mathcal{R}\\ m\neq k}} x_{ij}^{mks} \qquad \forall\, [i,j] \in Q,\ k \in \mathcal{R} \tag{4.13}$$

$$z_{ij}^s + \sum_{k\in\mathcal{R}}\sum_{l\in\mathcal{R}} x_{ij}^{kls} = 1 \qquad \forall\, [i,j] \in Q \tag{4.14}$$

$$x_{ij}^{kls} \leq h_k \qquad \forall\, [i,j] \in Q,\ \forall\, k,\ l \in \mathcal{R} \tag{4.15}$$

$$y_{ij}^{kls} \leq h_l \qquad \forall\, [i,j] \in Q,\ \forall\, k,\ l \in \mathcal{R},\ \ k \neq l \tag{4.16}$$

$$x_{ij}^{kls} \geq 0 \qquad \forall\, [i,j] \in Q,\ \forall\, k,\ l \in \mathcal{R} \tag{4.17}$$

$$y_{ij}^{kls} \geq 0 \qquad \forall\, [i,j] \in Q,\ \forall k,\ l \in \mathcal{R},\ k \neq l \tag{4.18}$$

$$z_{ij}^s \geq 0 \qquad \forall\, [i,j] \in Q \tag{4.19}$$

and $\mathbf{E}[\mathcal{Q}(h, \tilde{w})]$ which also can be shown by $\boldsymbol{Q}(h)$ is the expected value function offering the expectation of recourse with respect to random demand. The proposed recourse function satisfy the fixed and complete recourse assumptions.

4.2 Solution Approach

To solve the proposed model(SMnP), we devise an exact solution method based on BD algorithm widely known as the L-shaped method ([51]) in the context of stochastic programming. The BD algorithm decomposes the overall model into a master problem and a subproblem which are solved iteratively. In the stochastic programming context the master problem is considered as the first-stage problem (FSP) and subproblem as the second-stage problem (SSP). Although the BD is an effective solution method to solve MILP problems, it can not guarantee runtime efficiency to solve the problem. To address this issue, we suggest a number of enhancing schemes that can improve the performance of the BD. Among the reasons of runtime inefficiency, we can say that the BD does not have a procedure to solve the tie in selecting the best cut among a set of cuts generated in each iteration to be added to the master problem. In this study, we discuss and employ cut an strengthening technique to decrease the number of required BD optimalily cuts and solution time. Also, we offer two different types of disaggregation techniques to improve the bounds of algorithm faster. We suggest using the Progressive Hedging algorithm to generate initial information for L-shaped method, which as we show, is very effective approach. The mentioned methods with a number of unique enhancing techniques such as considering mean-value lower bounding and shortest path cuts are discussed in the current section. First, we explain the L-shaped method's main concepts.

4.2.1 L-shaped Primal/Dual Subproblem (PLSSP/DLSSP)

The L-shaped is a scenario decomposition technique which means it solves several smaller subproblem (for each scenario) and combines their results, instead of solving one huge subproblem for all scenarios. Given the locations of RPs, we can write the primal L-shaped subproblem or second stage primal problem for any $s \in \mathcal{S}$ as the model below.

$$\textbf{Min} \sum_{[i,j]\in Q} \sum_{k} \sum_{l} t_1(d_{ik} + d_{jl}) w_{ij}^{s} x_{ij}^{kls} + \sum_{[i,j]\in Q} \sum_{k} \sum_{l} t_2 d_{kl} w_{ij}^{s} y_{ij}^{kls} + \sum_{[i,j]\in Q} t_3 d_{ij} w_{ij}^{s} z_{ij}^{s} \quad (4.20)$$

subject to

$$(4.13), (4.14), (4.17), (4.18), \text{and } (4.19)$$

$$x_{ij}^{kls} \leq \hat{h}_k \qquad \forall\, [i,j] \in Q,\ \forall\, k,\ l \in \mathcal{R} \tag{4.21}$$

$$y_{ij}^{kls} \leq \hat{h}_l \qquad \forall\, [i,j] \in Q,\ \forall\, k,\ l \in \mathcal{R},\ \ k \neq l \tag{4.22}$$

where $\hat{h}_k$ is the first stage variable value for any $k \in \mathcal{R}$. Defining the dual variables ${q_1}_{ij}^{k}$, ${q_2}_{ij}$, ${q_3}_{ij}^{kl}$, and ${q_4}_{ij}^{kl}$ for constraints (4.13), (4.14), (4.21), and (4.22), respectively, we formulate the L-shaped dual subproblem or second stage dual problem for any $s \in \mathcal{S}$ as the following model.

$$\text{Max} \quad \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{k\in\mathcal{R}}\sum_{l\in\mathcal{R}}(\hat{h}_k {q_3}_{ij}^{kls} + \hat{h}_l {q_4}_{ij}^{kls}) + \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}} {q_2}_{ij}^{s} \tag{4.23}$$

subject to

$${q_1}_{ij}^{ks} - {q_1}_{ij}^{ls} + {q_2}_{ij}^{s} + {q_3}_{ij}^{kls} \leq t_1(d_{ik} + d_{lj})w_{ij}^{s} \qquad \forall\, [i,j] \in Q,\ \forall\, k,\ l \in \mathcal{R} \tag{4.24}$$

$${q_1}_{ij}^{ls} - {q_1}_{ij}^{ks} + {q_4}_{ij}^{kls} \leq t_2 d_{kl} w_{ij}^{s} \qquad \forall\, [i,j] \in Q,\ \forall\, k,\ l \in \mathcal{R},\ k \neq l \tag{4.25}$$

$${q_2}_{ij}^{s} \leq t_3 d_{ij} w_{ij}^{s} \qquad \forall\, [i,j] \in Q \tag{4.26}$$

$${q_3}_{ij}^{kls} \leq 0 \qquad \forall\, [i,j] \in Q,\ \forall\, k,\ l \in \mathcal{R} \tag{4.27}$$

$${q_4}_{ij}^{kls} \leq 0, \qquad \forall\, [i,j] \in Q,\ \forall\, k,\ l \in \mathcal{R},\ \ k \neq l. \tag{4.28}$$

4.2.2 L-shaped Master Problem (LSMP)

We formulate the master problem LSMP^m for any iteration m of the L-shaped method as the following model

$$\textbf{[LSMP]} \qquad \textbf{Min} \quad \sum_k f_k h_k + \tilde{\eta} \tag{4.29}$$

subject to

$$\text{BCutSet}(\tilde{\eta})^m \qquad \forall\, m \in \mathcal{O}^m \tag{4.30}$$

(4.11),

where the $\tilde{\eta}$ is a general term used to represent the auxiliary variable or a combination of auxiliary variables defined to facilitate the communication between L-shaped master and subproblem/s. Also the BCutSet$(\tilde{\eta})^m$ represents the Benders cut or cut set added at iteration m and $\mathcal{O}^m$ is the set of all L-shaped iterations from 1 to m. As it is obvious, the number and format of the optimality cut or cut set are the functions of $\tilde{\eta}$. We discuss the different versions of $\tilde{\eta}$ creating different versions of L-shaped method in Section 4.2.3.3.

4.2.3 Enhancing the L-shaped method

The L-shaped method is efficient in decomposing the overall problem into a set of smaller problems, solved effectively, but it cannot guarantee having a reasonable runtime for different problems. To have a desired runtime for our problem we suggest and implement a number of improving approaches which we show that they can enhance the performance of the L-shaped and reduce its runtime dramatically.

4.2.3.1 Strengthening the Benders' cuts

As we mentioned before, the subproblem, for any scenario, has network flow problem structure having degeneracy. Therefore, the DLSSP for any scenario has multiple optimal solutions, each of which provides a different cut with different strength in terms of cutting off the LSMP feasible area. This is important since by picking a proper optimal solution of DLSSP for any scenario in each iteration of the algorithm, we can generate stronger Benders' cut and accelerate the algorithm. As we stated earlier, for an optimization problem

$Min_{y \in Y, z \in R}\{z : f(u) + yg(u) \leq z, \; \forall u \in U\}$, [48] introduce the concept of strongness of a cut as follows. The cut $f(u_0) + yg(u_0) \leq z$ is stronger than the cut $f(u_1) + yg(u_1) \leq z$ if $f(u_0) + yg(u_0) \geq f(u_1) + yg(u_1) \; \forall y \in Y$ with a strict inequality for at least one $y \in Y$. Similar to our first work, but in a stochastic setting, to build a strengthened Benders cut, we use a technique similar to the two-phase approach presented by [49] and utilized by [50]. For any scenario of our problem, we can observe that the dual values associated with $\hat{h}_k$ with a zero value, in an iteration of the algorithm, do not have any contribution in the optimal objective value of DLSSP for any scenario of the problem. Hence, it is possible to modify the coefficient of a $\hat{h}_k = 0$ ($\hat{h}_l = 0$), that is ${q_3}_{ij}^{kls}$ (${q_4}_{ij}^{kls}$), without any effect on the DLSSP^s objective function value, for any scenario $s \in \mathcal{S}$. To do so, first we solve the reduced DLSSP^s to attain the dual variable values (${q_3}_{ij}^{kls}$ and ${q_4}_{ij}^{kls}$) associated with open RPs ($\hat{h}_k = 1$ and $\hat{h}_l = 1$), and the optimal values for ${q_1}_{ij}^{ks}$ and ${q_2}_{ij}^{s}$ $\forall\, [i,j] \in Q$, $\forall\, k \in \mathcal{R}$, and $\forall\, s \in \mathcal{S}$ as well. To generate these problems, we define $\mathcal{G}$ as the set of open RPs in each iteration, and by solving the following problems we can have the mentioned values.

$$[\mathit{Reduced}\,\mathbf{DLSSP}^s] \quad \text{Max} \quad \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} \sum_{k \in \mathcal{G}} \sum_{l \in \mathcal{G}} (\hat{h}_k {q_3}_{ij}^{kls} + \hat{h}_l {q_4}_{ij}^{kls}) + \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} {q_2}_{ij}^{s} \tag{4.31}$$

subject to

$$
\begin{aligned}
&{q_1}_{ij}^{ks} - {q_1}_{ij}^{ls} + {q_2}_{ij}^{s} + {q_3}_{ij}^{kls} \leq t_1(d_{ik} + d_{lj})w_{ij} && \forall\, [i,j] \in Q, \; \forall\, k, \; l \in \mathcal{G} && (4.32)\\
&{q_1}_{ij}^{ls} - {q_1}_{ij}^{ks} + {q_4}_{ij}^{kls} \leq t_2 d_{kl} w_{ij}, && \forall\, [i,j] \in Q, \; \forall\, k, \; l \in \mathcal{G}, \; k \neq l && (4.33)\\
&{q_2}_{ij}^{s} \leq t_3 d_{ij} w_{ij}, && \forall\, [i,j] \in Q && (4.34)\\
&{q_3}_{ij}^{kls} \leq 0 && \forall\, [i,j] \in Q, \; \forall\, k, \; l \in \mathcal{R} && (4.35)\\
&{q_4}_{ij}^{kls}, \leq 0, && \forall\, [i,j] \in Q, \; \forall\, k, \; l \in \mathcal{G}, \; k \neq l. && (4.36)
\end{aligned}
$$

Let $\mathcal{G}' = \mathcal{R} \backslash \mathcal{G}$. Then, in the second phase, solving the following model $\forall s \in \mathcal{S}$ results a set of dual variable values (associated with $\hat{h}_k = 0, \; \hat{h}_l = 0 \; \forall k, l \in \mathcal{G}'$) utilized to generate strengthened Benders cut.

$$\text{Max} \quad \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{k\in\mathcal{G}'}\sum_{l\in\mathcal{G}'}({q_3}_{ij}^{kls} + {q_4}_{ij}^{kls}) \tag{4.37}$$

subject to

$$\hat{q_1}_{ij}^{ks} - \hat{q_1}_{ij}^{ls} + \hat{q_2}_{ij}^{s} + {q_3}_{ij}^{kls} \leq t_1(d_{ik} + d_{lj})w_{ij} \qquad \forall\ [i,j]\ \in Q,\ \forall\ k,\ l \in \mathcal{G}' \tag{4.38}$$

$$\hat{q_1}_{ij}^{ls} - \hat{q_1}_{ij}^{ks} + {q_4}_{ij}^{kls} \leq t_2 d_{kl} w_{ij}, \qquad \forall\ [i,j]\ \in Q,\ \forall\ k,\ l \in \mathcal{G}',\ k \neq l \tag{4.39}$$

$${q_3}_{ij}^{kls} \leq 0, \qquad \forall\ [i,j]\ \in Q,\ \forall\ k,\ l \in \mathcal{G}',\ k \neq l \tag{4.40}$$

$${q_4}_{ij}^{kls} \leq 0, \qquad \forall\ [i,j]\ \in Q,\ \forall\ k,\ l \in \mathcal{G}',\ k \neq l. \tag{4.41}$$

The optimal solution of the second phase problem can be attained by using the following approach without actually solving it as a linear program. We can rewrite Constraint (4.38) as ${q_3}_{ij}^{kls} \leq \delta_{ij}^{kls}$ where $\delta_{ij}^{kls} = t_1(d_{ik}+d_{lj})w_{ij} - (\hat{q_1}_{ij}^{ks} - \hat{q_1}_{ij}^{ls} + \hat{q_2}_{ij}^{s})$, $\forall\ [i,j]\ \in Q,\ \forall\ k,\ l \in \mathcal{G}',\ \forall s \in \mathcal{S}$. Thus, the optimal value of ${q_3}_{ij}^{kls}$ can found using the following equation

$${q_3}_{ij}^{kls} = \begin{cases} \delta_{ij}^{kls} & \text{if } \ \delta_{ij}^{kls} < 0 \\ 0 & \text{otherwise} \end{cases} ,\ \forall\ [i,j]\ \in Q,\ \forall\ k,\ l \in \mathcal{G}'.$$

The optimal value of ${q_4}_{ij}^{kls}$ is achievable in a similar way described above utilizing Constraint (4.39).

4.2.3.2 Problem-based cut disaggregation

The subproblem is separable and we can separate the model to $|\mathcal{Q}|$ independent shortest path problems. This property enables us to generate more than one cut considered in each iteration of the L-shaped for each scenario. In this study, we use the ODReg cut disaggregation method introduced in previous chapter for RP-network design problem in deterministic environment, modified to be used in stochastic setting. We disaggregate Benders optimality cut based on origin and destination regions of the commodities. This disaggregation technique performs well in solving uncapacitated network problems.

ODReg cut disaggregation method works based on clustering the commodities regarding their origin and destination regions. To have this, we separate the geographical area into a number of equal-sized rectangles (regions) and generate one cut for commodities originating in and destined to the same regions(i.e., we generate one cut for all commodities originating in Region O and destined to Regions D). By defining REG_{OD} as the set of all region pair O and D that at least one commodity originates in O and destined to D, and C_{OD} as the set of commodities associated with region pair O and D. These cuts for single scenario for $s \in \mathcal{S}$ can be written as

$$\eta^s_{r_{OD}} \geq \sum_{[i,j]\in C_{OD}} \sum_{k\in\mathcal{R}} \sum_{l\in\mathcal{R}} (h_k {q_3}^{kls}_{ij} + h_l {q_4}^{kls}_{ij}) + \sum_{[i,j]\in C_{OD}} {q_2}^s_{ij} \qquad \forall\, r_{OD} \in REG_{OD}. \tag{4.42}$$

4.2.3.3 Scenario decomposition-based cut generation

The L-shaped method is one of the scenario decomposition methods which can decompose the second-stage problem to a set of smaller problem of size $|\mathcal{S}|$. Utilizing this property of L-shaped, we can generate different versions of it with different characteristics. In this section, we first introduce two basic versions of L-shaped method. Afterwards, we propose a scenario categorization scheme which can be used to build a new version of L-shaped.

The conventional single-cut version of L-shaped method suggests adding one cut in each iteration of the algorithm generated using a combination of all affine functions suggested by recourse function for each scenario. These affine functions are the same as the optimalily cut in the deterministic Benders decomposition framework. The form of this cut is as follows

$$\eta_{r_{OD}} \geq \sum_{s\in S} p^s(\sum_{[i,j]\in C_{OD}} \sum_{k} \sum_{l} h_k {\hat{q}_3}{}^{kls}_{ij} + h_l {\hat{q}_4}{}^{kls}_{ij} + \sum_{[i,j]\in C_{OD}} {\hat{q}_2}{}^s_{ij}) \qquad \forall\, r_{OD} \in REG_{OD} \tag{4.43}$$

which by replacing it by constraint set (4.29) in LSMP we can have the single-cut L-shaped method. In this case, the term $\tilde{\eta}$ in Function (4.29) should be replaced by $\sum_{r_{OD}\in REG_{OD}} \eta_{r_{OD}}$.

Having a unique affine function or optimality cut for each scenarios gives the opportunity of adding all of them to the master problem in each iteration. Doing so, we build the multi-cut version of L-shaped method. In this case, we should substitute the following constraint for Set (4.30).

$$\eta_{r_{OD}}^{s} \geq \sum_{[i,j]\in C_{OD}} \sum_{k} \sum_{l} h_k \hat{q}_{3ij}^{kls} + h_l \hat{q}_{4ij}^{kls} + \sum_{[i,j]\in C_{OD}} \hat{q}_{2ij}^{s} \qquad \forall\, r_{OD} \in REG_{OD}, \forall s \in \mathcal{S} \quad (4.44)$$

Also, the term $\tilde{\eta}$ in the SMP objective function should be changed to $\sum_{r_{OD}\in REG_{OD}} \sum_{s\in S} p^s \eta_{r_{OD}}^s$.

These two approaches have efficiencies and deficiencies. The multi-cut version by adding a large number of cuts in each iteration generate a tighter bound for the overall problem, because it has more information about the recourse problem. But adding $|\mathcal{S}|$ per iteration implies that the master problem size grows faster. This is more critical when a large number of scenarios should be considered. Hence, using this approach, in spite of having a lower number of iterations, we may have a longer runtime. On the other hand, adding one cut in each iteration although does not increase the size of master problem tremendously, the underlying master problem cannot generate strong lower bound in the initial iterations of the algorithm, and it typically needs more iterations than multi-cut version. A good trade-off between these two approaches can decrease the overall runtime.

In this study, we suggest a scenario categorization scheme which can be utilized to add more than one cut in each iteration to address the trade-off mentioned. In this approach we cluster the scenarios into three or more categories based on the their demand values. It can be done by defining three or more non-overlapping intervals and assign all the scenarios to them. It is possible to add one cut for each scenario category. We modify the master problem by defining $\mathcal{SC}$ as the set of scenarios categories and replacing the constraint set (4.30) by

$$\begin{aligned}\eta_{r_{OD}}^{c} \geq \sum_{s_c\in c} p^{s_c} (\sum_{[i,j]\in C_{OD}} \sum_{k} \sum_{l} h_k \hat{q}_{3ij}^{kls_c} + h_l \hat{q}_{4ij}^{kls_c} \\ + \sum_{[i,j]\in C_{OD}} \hat{q}_{2ij}^{s_c}) \qquad \forall\, r_{OD} \in REG_{OD}, \forall c \in \mathcal{SC} \qquad (4.45)\end{aligned}$$

and changing the term $\tilde{\eta}$ to $\sum_{r_{OD}\in REG_{OD}} \sum_{c\in \mathcal{SC}} p^c \eta_{r_{OD}}^c$, where p^{s_c} is the probability of scenario s in the category $c \in \mathcal{SC}$. We show that adding these scenario category-based (SC-based) cuts can improve the performance of the L-shaped method for our problem.

Considering this type of cut for the L-shaped method forms a new version of L-shaped which we call it "SC-based cut L-shaped".

4.2.3.4 Mean value lower bounding cut

Based on Jensen inequality, it can be easily shown that for any convex recourse function the mean value scenario can generate a valid lower bound for the recourse for any feasible first stage solution. This concept can be used to develop a cut called mean value lower bounding cut which can be added to the BMP in each iteration of L-shaped to tighten the lower bound of the problem ([52]; [53]).

Proposition 1: Suppose $\hat{h}$ and $\mathcal{Q}(\hat{h}, \bar{\omega})$ represent a feasible first stage solution and corresponding recourse value for mean value scenario and $\bar{\omega^c}$ is the mean value scenario for scenario category c, the following inequalities are valid

$$\eta \geq \mathcal{Q}(\hat{h}, \bar{\omega}), \tag{4.46}$$

$$\sum_{s \in S} p^s \eta^s \geq \mathcal{Q}(\hat{h}, \bar{\omega}), \tag{4.47}$$

$$\eta^c \geq \mathcal{Q}(\hat{h}, \bar{\omega^c}), \qquad \forall c \in \mathcal{SC} \tag{4.48}$$

for single-cut, multi-cut and category-cut L-shaped, respectively.

Proof. Our problem has complete recourse and the recourse function $\mathcal{Q}$ is bounded. According to Jensen inequality ([54]) we can write

$$\boldsymbol{\mathcal{Q}}(\hat{h}) \geq \mathcal{Q}(\hat{h}, \bar{\omega}) \tag{4.49}$$

$$\boldsymbol{\mathcal{Q}}^c(\hat{h}) \geq \mathcal{Q}(\hat{h}, \bar{\omega^c}) \qquad \forall c \in \mathcal{SC} \tag{4.50}$$

where $\boldsymbol{\mathcal{Q}}^c(\hat{h})$ is expected recourse function for scenario category c. Also, regarding our discussion in Section 4.2.3.3, for any first-stage solution $\hat{h}$ we can write the following inequalities

(in the closed format used above)

$$\eta \geq \boldsymbol{\mathcal{Q}}(\hat{h}) \tag{4.51}$$

$$\sum_{s \in S} p^s \eta^s \geq \boldsymbol{\mathcal{Q}}(\hat{h}) \tag{4.52}$$

$$\eta^c \geq \boldsymbol{\mathcal{Q}}^c(\hat{h}), \qquad \forall c \in \mathcal{SC}. \tag{4.53}$$

The Inequality (4.52) is attained by taking expectation of the both sides of the $\eta^s \geq \mathcal{Q}(\hat{h}, w^s)$ over all the scenarios in $\mathcal{S}$. Therefore, (4.46) follows from (4.49) and (4.51), the (4.47) follows from (4.49) and (4.52), and (4.48) is deduced from (4.50) and (4.53). We show the effect of adding these cuts to the problem for three different versions of L-shaped method.

4.2.3.5 Early termination of LSMP

In the conventional L-shaped implementations, the master problem is solved to optimality for all the iterations of the algorithm. For some problems, reaching to an optimal solution is time-consuming, since the convergence rate of the IP solution approaches gets slower by getting closer to optimality, specially for large-size master problems. In this study, similar to our suggestion in the previous chapter, we suggest a step-wise LSMP early termination approach which can mitigate the long LSMP solution time issue in the L-shaped implementation. Using this approach, we terminate the LSMP solution process upon reaching to a predetermined optimality gap level. In this case, the lower bound of the LSMP is considered as the lower bound and the last incumbent solution found is used to calculate the upper bound for L-shaped. This approach can expedite the overall L-shaped algorithm dramatically by adding the Benders cuts faster, although the quality of these cuts may be lower than the original ones.

We perform this approach in a step-wise manner in such a way that we consider different levels of acceptable gap percentage for different iterations of LSMP. To do so, we start the algorithm with a large value of this level and we decrease it gradually to a smaller value for the later iterations. In this way, the optimality cuts can be added faster in the initial iterations of the L-shaped and the lower bound is increased sooner. Based on our experiments, we suggest this gap level percentage equal to 9% for initial iteration of L-shaped and 4%, 3%, 2%, 1% for second, third, fourth, fifth iterations, respectively, and finally 0.5% for sixth and onward.

4.2.4 A New Progressive Hedging (PH) for SMIP

In this section, we present a new version of Progressive Hedging (PH) to warm-start our L-shaped method and improve its performance. If we solve the single scenario problem for scenario s, we will get a $\hat{\mathbf{h}}$ vector (the RP location decision vector) which is unique for this specific scenario. The single scenario problem for any scenario $s \in \mathcal{S}$ can be written as follows:

$$\textbf{[MnP}^s\textbf{]} \qquad \textbf{Min} \sum_k f_k h_k + \sum_{[i,j]\in Q} \sum_k \sum_l t_1(d_{ik} + d_{jl}) w_{ij}^s x_{ij}^{kls} + \sum_{[i,j]\in Q} \sum_k \sum_l t_2 d_{kl} w_{ij}^s y_{ij}^{kls} + \sum_{[i,j]\in Q} t_3 d_{ij} w_{ij}^s z_{ij}^s \tag{4.54}$$

subject to

(4.11)

(4.13) - (4.19)

We write the *scenario* formulation (**ScP**)of our problem as the follows:

$$\textbf{[ScP]} \quad \textbf{Min} \sum_{s\in\mathcal{S}} p^s \Bigg(\sum_k f_k h_k^s + \sum_{[i,j]\in Q} \sum_k \sum_l t_1(d_{ik} + d_{jl}) w_{ij}^s x_{ij}^{kls} + \sum_{[i,j]\in Q} \sum_k \sum_l t_2 d_{kl} w_{ij}^s y_{ij}^{kls} + \sum_{[i,j]\in Q} t_3 d_{ij} w_{ij}^s z_{ij}^s \Bigg) \tag{4.55}$$

subject to

(4.2) - (4.8)

$$h_k^s = \bar{h}_k \qquad \forall\, k \in \mathcal{R}, \forall s \in \mathcal{S} \tag{4.56}$$

$$h_k^s \in \{0,1\} \qquad \forall\, k \in \mathcal{R}, \forall s \in \mathcal{S}. \tag{4.57}$$

Let $\mathbf{fh}^s$ and $\mathbf{g}^s\mathbf{y}^s$ represent RPs activation cost and total flow transportation cost for scenario $s \in \mathcal{S}$, respectively, and the associated feasible solutions set be denoted as $\mathcal{X}_s$. Hence, **ScP** can alternatively be shown as the following model.

$$\textbf{Min} \quad \sum_s p^s(\mathbf{fh}^s + \mathbf{g}^s\mathbf{y}^s) \tag{4.58}$$

subject to

$$(\mathbf{h}^s, \mathbf{y}^s) \in \mathcal{X}^s \qquad \forall\, s \in \mathcal{S} \tag{4.59}$$

$$\mathbf{h}^s = \bar{\mathbf{h}} \qquad \forall\, s \in \mathcal{S}. \tag{4.60}$$

Constraints (4.60) ((4.56)), ensuring equality of all the first-stage solutions ($\hat{\mathbf{h}}^s$), are called non-anticipativity constraints. Relaxing these constraints using Lagrangian multipliers enables us to decompose overall problem by scenarios. The Progressive Hedging (PH) algorithm presented by [38] is one of the scenario decomposition algorithms which works based on this technique and tries to satisfy the non-anticipativity constraints by finding a feasible $\bar{\mathbf{h}}$ iteratively. The PH algorithm, which can be categorized as one of the augmented Lagrangian methods, was originally proposed for stochastic linear programs (SLP) and it can be proved that this algorithm converges to an optimal solution for these type of problems. However, the convergence of PH for class of stochastic mixed-integer programs (SMIP) is not proved, although a number of algorithmic enhancements have been presented for PH. Nevertheless, the PH can be used as a strong heuristic for SMIP ([41]; [44]; [40]; [39]). In this study, we use this concept to build a PH-based algorithm whose solution provides a reliable initial solution for our L-shaped algorithm. Algorithm 4.2.4 presents the overall steps of the proposed PH.

Algorithm 4 Progressive Hedging(PH)

Initialize $\mathcal{SN}$, σ, $PHGLvl$ and $MaxK$
for $s \in \mathcal{SN}$ **do**
  $\hat{\mathbf{h}}^s_{\mathcal{K}} \in \mathbf{argmin}_{\mathbf{h}^s,\mathbf{y}^s}\{\mathbf{MnP}^s\}$
end for
Calculate PH_{LB}
Generate integer $\bar{\mathbf{h}}_{\mathcal{K}}$
Calculate PH_{UB} and $PH_{Gap}\%$
If $PH_{Gap}\% \leq PHGLvl$, terminate
$\mathcal{K} = \mathcal{K} + 1$
while $(\mathcal{K} \leq \texttt{MaxK}$ **and** $PH_{Gap}\% > \epsilon)$ **do**
  $\boldsymbol{\lambda}^s_{\mathcal{K}} = \boldsymbol{\lambda}^s_{\mathcal{K}-1} + \sigma(\mathbf{h}^s_{\mathcal{K}} - \bar{\mathbf{h}}_{\mathcal{K}})$
  for $s \in \mathcal{SN}$ **do**
    $(\hat{\mathbf{h}}^s_{\mathcal{K}}, \hat{\mathbf{y}}^s_{\mathcal{K}}) \in \mathbf{argmin}_{\mathbf{h},\mathbf{y}}\left\{\mathbf{fh} + \mathbf{g}^s\mathbf{y} + \boldsymbol{\lambda}^s_{\mathcal{K}}(\mathbf{h} - \bar{\mathbf{h}}_{\mathcal{K}}) + \frac{\sigma}{2}\left\|\mathbf{h} - \bar{\mathbf{h}}_{\mathcal{K}}\right\|^2 \middle| (\mathbf{h}, \mathbf{y}) \in \mathcal{X}^s\right\}$
  end for
  Calculate PH_{LB}
  Generate integer $\bar{\mathbf{h}}_{\mathcal{K}}$
  Calculate PH_{UB} and $PH_{Gap}\%$
  $\mathcal{K} = \mathcal{K} + 1$
end while
Return $\bar{\mathbf{h}}_{\mathcal{K}}$

The algorithm starts with initializing the associated parameters and constants. One of the parameters is $\mathcal{SN}$ which should be determined properly.

Determining $\mathcal{SN}$: The proposed PH can be used to solve the overall problem, as a viable heuristic, if we consider $\mathcal{SN} = \mathcal{S}$. However since the convergence of the PH in a reasonable amount of time cannot be guaranteed, we suggest solving the PH for a special set of constructed scenarios. Using this approach we can generate interesting first-stage decisions which we use them to start the L-shaped with a interestingly small initial optimality gap percentage. As we discussed in Section 4.2.3.3, we assign each scenario to a unique scenarios category and we have a set of scenario categories instead of a single scenario set. In this way, it is possible to create a mean-scenario for each category by calculating the expected value of the demand of each commodity over all the scenarios in that the category. We suggest using the proposed PH for a set containing these mean-scenarios and initialize the $\mathcal{SN}$ by this set. In the computational results section, we show that implementing our PH having the $\mathcal{SN}$ can result in a reliable $\bar{\mathbf{h}}_k$ (final first-stage decision vector) with a reasonable runtime.

After inializing the input parameters and constants, a single scenario problem for all $s \in \mathcal{SN}$ is solved resulting the $\hat{\mathbf{h}}^s_{\mathcal{K}}$ (the first-stage solution vector for each scenario) for all $s \in \mathcal{SN}$ at iteration $\mathcal{K} = 0$. We calculate the initial lower bound of the PH in Step (5). To do so, we solve the single scenario problems ($\mathbf{MnP}^s$s) utilizing the strengthened Benders decomposition method utilizing ODReg cut type. We select the ODReg cut type as the cut disaggregation method used.

Calculating initial lower bound for PH: Let OFV^s represents the objective function value of $\mathbf{MnP}^s$ fro any $s \in \mathcal{SN}$, then we have:

$$PH_{LB} = \sum_{s} p^s OFV^s \tag{4.61}$$

where PH_{LB} is the lower bound of PH.

Generating integer $\bar{\mathbf{h}}_{\mathcal{K}}$: The original PH algorithm for stochastic linear problems suggests the

$$\bar{\mathbf{h}}_{\mathcal{K}} = \sum_{s \in \mathcal{SN}} p^s \, \hat{\mathbf{h}}^s_{\mathcal{K}}$$

for all iterations $\mathcal{K}$s, but it cannot guarantee the integrality of $\bar{\mathbf{h}}_{\mathcal{K}}$. In the proposed version of the PH which can be used as a powerful heuristic to solve SMIPs, we suggest a heuristic approach to generating an integer vector $\bar{\mathbf{h}}_{\mathcal{K}}$, such that, first we calculate the $\bar{\mathbf{h}}_{\mathcal{K}}$ as offered by original PH, then we suggest using the following correction,

$$\bar{h}_{k\mathcal{K}} = \begin{cases} 1 & \text{if } \hat{h}^s_{k\mathcal{K}} = 1, \ \forall s \in \mathcal{SN} \\ 0 & \text{otherwise} \end{cases}, \ \forall \, k \in \mathcal{R},$$

to make the resulted $\bar{\mathbf{h}}_{\mathcal{K}}$ an integer vector.

Calculating upper bound for PH: We can calculate an upper bound for PH by evaluating any integer h (or generating the expected result of implementing h) using following equation,

$$PH_{UB} = \mathbf{f}\bar{\mathbf{h}} + \mathbf{E}[\mathcal{Q}(\bar{\mathbf{h}}, \tilde{w})].$$

The validity of this upper bound can be easily shown by observing that any integer first-stage decision vector can generate a valid upper bound for **SMnP**.

Calculating optimality gap percentage for PH: We calculate the optimality gap percentage for PH as follows.

$$PH_{Gap}\% = \frac{PH_{UB} - PH_{LB}}{PH_{UB}} \times 100$$

The gap percentage ($PH_{Gap}\%$) should be used to examine the associated stopping criterion at Step 8. It should be noted that PH_{LB}, PH_{UB} and $PH_{Gap}\%$ can be considered as the global lower bound, upper bound and gap percentage for overall SMIP, respectively, if we use the proposed PH for $\mathcal{SN}$ equal to $\mathcal{S}$.

The PH algorithm main loop is started at Step 10 after updating the iteration index $\mathcal{K}$ at Step (9). Updating the Lagrangian dual decisions is the first step of the main loop. The equation presented in Step (11) is used to do so, where the σ represents the step-size. Similar to other gradient methods, σ plays an important role in the convergence of the PH, although its convergence cannot be guaranteed in finite number of iterations. The main loop is continued with solving a set of scenario problems formed by decomposing **ScP** into $|\mathcal{S}|$ scenario problems. As we mentioned earlier, PH algorithm is in the class of augmented Lagrangian methods which augment the Lagrangian with a proximal (regularization) term in objective function of the problem. Hence, in any iteration $\mathcal{K} > 0$ the objective function of the mentioned scenario problems has two additional terms, a penalty term containing the dual prices for penalizing the non-anticipativity constraint relaxation (the $\boldsymbol{\lambda}^s_{\mathcal{K}}$ represents the Lagrangian multiplier vector for scenario s at iteration $\mathcal{K}$) and a proximity term controlling the difference between the scenario solution and $\bar{\mathbf{h}}$. Similar to initial iteration of the algorithm we utilize the the strengthened Benders decomposition method utilizing ODReg cut type presented in the previous chapter to solve the discussed scenario problems.

Calculating lower bound for PH: To calculate the lower bound for PH, we use

Equation (4.61) by changing the OFV^s at any iteration k as follows:

$$OFV^s = \mathbf{f}\hat{\mathbf{h}}^s_{\mathcal{K}} + \mathbf{g}^s\hat{\mathbf{y}}^s_{\mathcal{K}} \tag{4.62}$$

where the $\hat{\mathbf{h}}^s_{\mathcal{K}}$, $\hat{\mathbf{y}}^s_{\mathcal{K}}$ are the solution of the problems solved in Step (13). [43] show that the presented lower bound is valid at any iteration $\mathcal{K}$ of PH algorithm.

Steps (16) and (17) are done exactly as (6) and (8), respectively. Two stopping criteria considered are the gap percentage and the number of iterations. The proposed PH is terminated either when the gap percentage is less than ϵ or $\mathcal{K}$ is greater than $MaxK$. Finally, the resulted $\bar{\mathbf{h}}_{\mathcal{K}}$ is stored to be used as the initial solution of the L-shaped method.

4.2.5 Progressive Hedging-Strengthened L-Shaped (PH-SLS) Implementation

Algorithm 5 presents the overall steps of the proposed Progressive Hedging-Strengthened L-Shaped (PH-SLS). We initialize the algorithm's parameters and constants in Step 1. Having the first-stage decisions, we can consider the second-stage problems as a set of shortest path problems. Hence, we solve the second-stage problem for all the scenarios by Dijkstra's algorithm by assuming that all the RPs are available to be used (free of charge). The Dijkstra's algorithm, assuming the length of each arc is equal to its actual unit shipment cost, returns the shortest paths information for all commodities' flows and all scenarios. Note that, as we discussed, the unit cost of the arcs are determined based on the arc type (network or direct shipment arc type). The results of the Dijkstra are used in the next two steps of the algorithm. We use the resulted shortest paths information to run the strengthened Benders decomposition (the overall algorithm presented in the previous chapter) embedded in the proposed PH algorithm (Algorithm 4). We add a set of shortest path cuts (SPC) to the L-shaped master problem (LSMP). By defining $SPCost^s_{ij}$ as the shortest path cost of commodity $[i,j]$ of scenario s, $\forall\ [i,j] \in \mathcal{Q}$, $\forall\ s \in \mathcal{S}$, Equation (4.63) represents the form of these cuts for the SC-based cut version of L-shaped method.

$$\eta^c_{r_{OD}} \geq \sum_{s_c \in c} \sum_{[i,j] \in C_{OD}} p^{s_c} SPCost_{ij} \qquad \forall\ r_{OD} \in REG_{OD},\ \forall\ c \in \mathcal{SC} \tag{4.63}$$

Shortest path value for each commodity in each scenario is a lower bound on the transportation cost of that commodity.

An initial upper bound for the L-shaped can be calculated based on the shortest paths information. To do so, we first detect the RPs that have not been used in any of shortest paths found for all the flows and scenarios, as the unused RPs. Hence, we set the first-stage decision vector $\hat{\mathbf{h}}$ by giving zero to $\hat{\mathbf{h}}_k$ for which corresponding RP is a unused RP and one to the others. Now we can calculate the recourse for each scenario and expected recourse using the shortest paths values. Calculating the total activating cost of locating the used RPs recently found and adding it to the expected recourse, we can have an initial upper bond for the problem.

Algorithm 4 is employed in Step 5 and the resulted $\bar{\mathbf{h}}_{\mathcal{K}}$ is used as initial RP location decisions to start L-shaped main loop starting at step 6. Using these decisions, we solve the dual L-shaped subproblem (DLSSP) for each scenario and calculate the upper bound (UB) for the problem. Upper bound for L-shaped is calculated by using Equation (4.64) in which the OFV^s represents the DLSSP objective function value of scenario s.

$$UB = \min\{UB,\ \mathbf{f}\hat{\mathbf{h}} + \sum_{s} p^s OFV^s\} \tag{4.64}$$

The algorithm run time (`runtime`) and optimality gap (`optgap`) are updated in step (8). The Benders' cut strengthening procedure described in Section 4.2.3.1 is performed in step 9 and the proposed optimality cuts having SC-based, disaggregated and strengthened properties are generated and added to the LSMP. The MVP cut set discussed in Section 4.2.3.4 is added the LSMP in step 11. The master problem is solved in step 12. Since we consider early termination method discussed in Section 4.2.3.5, the lower bound (LB) of LSMP is considered as the new lower bound candidate. Similar to UB updating process, we always keep the best LB attained as the global lower bound of the algorithm. Step 13 is done similar to Step 8 and if the stopping criteria are not met, we update the $\hat{\mathbf{h}}$. Finally, the RP location decisions are reported as the final first-stage solution.

Algorithm 5 Progressive Hedging-Strengthened L-Shaped (PH-SLS)

1: **initialize** $\epsilon = 0.03$, `Optgap` $= 1.0$, `Runtime`$=0$, `Stoptime`$=20000$
2: Solve all the single scenario problems using Dijkstra's algorithm assuming $f_k = 0;\ \forall k \in \mathcal{R}$
3: Add shortest path cuts to the L-shaped MP
4: Calculate UB
5: Employ Algorithm (4)
6: **while** (`Runtime` $\leq$ `Stoptime` **and** `optgap` $> \epsilon$) **do**
7: Solve DLSSP, $\forall\ s \in \mathcal{S}$, and calculate the Upper Bound
8: Update `Runtime` and `optgap`
9: Update the DLSSPs variables values for strengthened Benders cuts
10: Generate SC-based-disaggregated-strengthened Benders cuts and add them to LSMP
11: Add MVP cut set
12: Solve **LSMP** and update the Lower Bound
13: Update `Runtime` and `optgap`
14: Update the $\hat{\mathbf{h}}$ by using the results of the recently solved **LSMP**
15: **end while**
16: **return** $\hat{\mathbf{h}}$

4.3 Computational Study on Algorithmic Performance

In this section we first describe our approach to generate testbeds be used in evaluating all the aspects of the proposed model and solution methodology. Afterwards, we report and discuss a comprehensive numerical study on the test instances generated.

4.3.1 Data Generation

We generate six classes of random test instances to evaluate the performance of the proposed solution method. We generate $|\mathcal{N}|$ clustered random point coordinates in a rectangle resembling origin and destinations nodes and a geographical region, respectively. The dimension of the rectangular is 150×100 (width $\times$ height). Also, we generate $|\mathcal{R}|$ uniformly distributed point coordinates in the mentioned rectangle to consider a set of RP location candidates. To consider the point clustering concept mentioned above, we divide the whole rectangular by 25 sub-rectangular. Afterwards, we randomly generate 75% of the points in five of them selected randomly. Also, to maintain the diversity among the points, the remaining 25% are generated all over the main rectangular uniformly. Similarly, we generate the potential RP locations in the main area uniformly.

The specifications of the generated test classes are presented in Table 4.2.

Table 4.2: The characteristics of different test problem classes

Class	$\|\mathcal{N}\|$	$\|\mathcal{R}\|$	$\|\mathcal{Q}\|$	f
C1	120	25	4998	405000
C2	120	30	4998	405000
C3	140	30	6811	540000
C4	140	35	6811	540000
C5	160	35	8904	690000
C6	160	40	8904	690000

The first column of this table show the index of each class. The second, third and forth columns present the number of origin/destination nodes, RP locations and commodities, respectively. Finally, the fifth column reports the fixed-cost of activation an RP location. The increment considered in this column is because of reflecting this fact that to serve a higher number of TLs we need more expensive facilities.

The number of commodities for each class $|\mathcal{Q}|$ is equal to $|\mathcal{N}|(|\mathcal{N}|-1)D$, where D is the density parameter representing the percentage of node (origin/destination) pairs having positive flow among all the available node pairs. The default value for D is 35%.

To have more realistic test instances, we perform the following procedure. Based on proximity, we categorize the commodities into three groups, long, medium, and short range commodities. To specify these three category, we use the double (th_1, th_2) containing two thresholds th_1 and th_2 to separate short-medium-long distances such that any commodity having distance less than th_1 is considered as short-distance, between th_1 and th_2 as medium-distance and greater than th_2 as long-distance commodity. Afterward, to select different commodities we define the triple (L, M, S) which its first, second and third element represents the percentage of short, medium and long-distance commodities in the instances. $S\%|\mathcal{Q}|$, $M\%|\mathcal{Q}|$, and $L\%|\mathcal{Q}|$ different commodities are randomly picked from the short, medium, and long commodity groups, respectively. Based on the size of geographical region considered, we set (th_1, th_2)=$(30, 60)$ and (L, M, S)=(80, 10, 10) to generate all the test instances.

To consider the uncertainty in the demand of the commodities (w_{ij}) we divide all the scenarios into three equal-size groups, low, medium and high-demand scenarios. The demands of the low-group scenarios are generated by using uniform distribution U[50, 100], medium-group by U[125, 175] and high-group by U[200, 250] based on the number of TLs. In this study, the number of scenarios in all scenario groups are equal.

The unit transportation cost values t_1, t_2 and t_3 are set to 1, 2 and 3, respectively. We variate the unit transportation cost for different types of transportation, considered, to reflect this fact that the driving for longer distances imposes implicit costs to the system mostly because of driver turn over issues. Finally, the default values for double (Δ_1, Δ_2) are considered as $(20, 40)$ for all test instances.

4.4 Numerical Results

Before solving the test instances by the proposed algorithm, we perform a number of experiments to evaluate the performance of different parts of the suggested solution method. Note that in all the tables following "Class" shows the test instances classes, “Gap%” reports the optimality gap percentage and “T(s)” reports the runtime in seconds attained by the considered algorithm after meeting any of the stopping criteria. We solve five test instances of each class and report the average of results.

Note that, based on practice, we set $\sigma = 100$, $PHGLvl = 3\%$ and $MaxK = 2$ in any implementation of PH algorithm presented as Algorithm (4). To solve all the presented models, we use CPLEX 12.6.1, and all of the experiments are conducted using C++ on machines with an Intel Core i7-4790 CPU at 3.6 GHz, 32 GB RAM running 64-bit OS.

4.4.1 Single-Multi-SC-based L-shaped Algorithm

To discuss the effectiveness of the SC-based cuts to solve the problem, we compare the results of two well-known versions of L-shaped to the version that we suggest by adding the SC-based cuts in each iteration. Table 4.3 shows the results of the three mentioned approaches. Also, this table shows the results of these approaches considering mean-value cut set which we discuss them later. It should be noted that to implement these methods we use the problem-based ODReg cut disaggregation method described in Section 4.2.3.2 and we do not include any heuristic to improve the overall performance of them to have an exact comparison between the approaches.

Table 4.3: The results of single, multi and SC-based cut L-shaped methods with/without MVP

Class	L-Shaped_Scut				L-Shaped_SCbcut				L-Shaped_Mcut			
	without MVP		with MVP		without MVP		with MVP		without MVP		with MVP	
	Gap%	T(s)	Gap%	T(s)	Gap%	T(s)	Gap%	T(s)	Gap%	T(s)	Gap%	T(s)
C3	2.28	6460.4	2.49	5665.6	1.87	6342.2	2.18	5400.8	4.616	20677.4	2.48	15709.0
C4	2.65	10992.6	2.59	10175.6	2.14	10662.8	2.48	8460.0	13.272	22663.8	6.71	23111.6
C5	2.43	14723.0	3.48	14729.8	2.52	12350.8	2.74	11763.2	14.762	24363.2	5.83	23873.3
C6	8.89	21172.2	8.74	20839.8	3.18	18217.2	3.32	16998.6	33.238	27032.6	15.36	21114.0

First, we compare the results of single and multi-cut approaches. Among these two, the single-cut version performs better such that the average runtime of the single-cut for all test classes is 13337.1 seconds which is about 56% of that of multi-cut version. Also, the average gap percentage of single and multi-cut are 4.1% and 16.5%, respectively. The multi-cut version cannot solve four of test instances of C3, four of C4, five of C5 and five of C6 upon meeting the stopping criteria of the algorithm whereas these values for single-cut version are zero, zero, one five, respectively. It is obvious that the multi-cut version does not have a good performance and we do not suggest using it to solve the considered problem. Although the single-cut version outperforms the multi-cut, compared to SC-based-cut version, suggested in this study, it performs weaker. The average runtime for SC-based-cut is 11893 which is about 89% of that of single-cut. Also, the average of gap% is 2.7% that is 34% less that this amount of single-cut. The SC-based-cut version cannot solve zero of test instances of C3, zero of C4 and C5, and two of C6. These amounts show that the usefulness of using SC-based-cut is more obvious in solving the large-size test instances, this is a favorable feature of this method. Therefore, we use this version of L-shaped method for our further computational studies conducted using Algorithm 5. It should be noted that in the our experiments, we start the algorithms with all closed RP locations for the sake of simplicity.

4.4.2 Adding MVP cut to SC-based L-shaped Algorithm

Table 4.3 also includes the results of adding MVP cut set in our L-shaped algorithm. Hence, we can evaluate the results of considering each of these cut types. Adding the MVP cut for multi-cut version, described in Equation (4.47), can improve the average runtime and gap% of the multi-cut L-shaped by about 12% and 54%, this is interesting, but it cannot make the multi-cut version efficient enough to be suggested for solving our problem. Utilizing MVP cut for single-cut L-shaped, presented in Equation (4.46), can improve the runtime about 4%, but cannot enhance the average gap%. Finally, adding MVP cut to the SC-based L-shaped algorithm can improve the runtime of it by 10% in an acceptable level of gap%(which is $\leq 3\%$). Furthermore, adding this cut, we can solve one of the two test instances of C6, which are not solvable by this algorithm without using MVP cuts. Therefore, we suggest adding MVP cut, proposed as Equation (4.48) to SC-based L-shaped algorithm. In conclusion we consider the SC-based L-shaped method with MVP cuts having the best overall performance among six different method discussed in Table 4.3, as the main approach to implement L-shaped method presented as Algorithm 5.

4.4.3 Adding SPC and PH into SC-based L-shaped Algorithm

Now, we show that considering both shortest path cuts and PH in the proposed solution procedure can improve the algorithm performance. To do so, we compare four different versions of the L-shaped with MVP method. As we discussed about the results, showed in Table 4.3, the SC-based cut L-shaped with MVP cut method have the best performance among six different versions of L-shaped. Hence, we select this approach to compare four different versions of it to show how embedding both SPC and PH can improve the performance of this algorithm. Table 4.4 presents the results of these implementations. Three of these four algorithms built based on the SC-based cut L-shaped method, shown as LS1, LS2, attained by adding PH, and LS3, formed by adding SPC to LS1 . Also, the forth method is PH-SLS, which is formed by embedding both SPC and PH into LS1, presented as Algorithm 5. We compare these four approached based on four measures, the gap%, the runtime (T(s)), the number of iterations done by L-shaped, showed as "No. It." and the gap of first iteration of L-shaped presented as "LSh. I. G%".

The average of Gap% of all four methods are less than 3% meaning that, on average, all the methods can solve the instances in the time limit which is 20000 seconds. LS2, LS3 and PH-SLS can solve all the instances, but one the instances of the C6 cannot be solved by LS1.

In terms of the runtime, LS2 can improve the runtime of LS1 by 14% and LS3 can do it by 3.5%. However, adding both of them to LS1, we build PH-SLS which can improve the runtime of LS1 by 25% which is impressive. Another important factor of our implementations is the initial gap% of the L-shaped part of the algorithms. LS1 on average starts with 97.3%, LS2 with 61.4%, and LS3 with 8.4%. Both SPC and PH are effective to decrease the initial gap% of the LS1 by improving both initial lower and upper bound of the algorithm. In this regard, SPC is more effective to improve the initial lower bound and PH is more effective to lower the initial upper bound of the algorithm. Taking advantage of adding both SPC and PH to LS1 (forming PH-SLS), we can start the L-shaped by initial gap% about 5.6%, on average.

The number of required L-shaped iterations is another important factor playing an important role in the amount of runtime. The average number of L-shaped iterations for LS1 is equal to 7.9, which this value for LS2 and LS3 are 5.3 and 6.2. The average number of L-shaped iterations is 3.8 which is less than half of that of LS1.

To conclude, we suggest PH-SLS to solve the problem as it can solve all the considered test instances in the shortest runtime attained. Moreover, to have a comparison between the results of PH-LSL and the single-cut L-shaped method, as a well-known method to solve the 2-SMIPs whose results are showed in Table 4.3, we should say that, on average, the PH-SLS can solve all the considered test instances (C3, C4, C5 and C6) in about 60% of required runtime for single-cut L-shaped method. The average gap% of PH-SLS is about 63% of single-cut L-shaped method average gap%. These differences are more prominent for two larger test classes C5 and C6, such that the average runtime of the PH-SLS for these two classes is about 53% and average gap% is about 44% of those of single-cut L-shaped method, respectively. These results imply that the difference between the performances of PH-SLS and S-cut L-shaped method increases by increasing size of the problem, making the PH-SLS a more attractive approach to solve large-size instances.

Table 4.4: The results of embedding SPC and PH

Class	SC-based cut L-shaped with MVP												PH-SLS			
	LS1: without SPC and PH				LS2: with PH without SPC				LS3: with SPC without PH							
	Gap %	T(s)	LSh. I. G%	No. It.	Gap %	T(s)	LSh. I. G%	No. It.	Gap %	T(s)	LSh. I. G%	No. It.	Gap %	T(s)	LSh. I. G%	No. It.
C3	2.18	5400.8	97.28	6.6	2.30	4725.4	35.57	4.2	2.09	5969.8	7.14	6	2.57	4563.2	6.04	3.2
C4	2.48	8460.0	97.57	8.2	2.48	8081.6	60.89	5.8	2.30	9947.4	9.11	6.4	2.77	8325.6	6.18	5.0
C5	2.74	11763.2	96.69	7.8	2.53	10040.8	71.86	5.2	2.53	10368.2	8.03	5.8	2.35	8790.2	6.15	3.6
C6	3.32	16998.6	97.49	9.0	2.73	13844.8	77.18	5.8	2.71	14872.6	9.29	6.6	2.62	10383.2	4.03	3.4

4.4.4 PH-SLS overall results

Table 4.5 shows the results of solving the six test classes by PH-SLS (Algorithm 5). The third, forth and fifth columns of this table report the average runtime of three major parts of the PH-SLS which are L-shaped, PH and SPC, respectively. The sixth and seventh columns present the average gap% and number of performed iteration of PH before starting the main L-shape loop. The gap% of PH is not global and it is just used to check its associated stopping criterion. Finally, the last column of the table shows the average number of RPs used to solve the instances.

The average gap% resulted by PH-SLS is 2.57% and all the test instances are solvable by this algorithm such that the average runtime attained is 6119.77 seconds. The interesting point about PH-SLS is that the rate of increasing runtime by increasing the size of the

Table 4.5: The results of solving six test classes by PH-SLS

Class	Gap%	T(s)	L-Shaped T(s)	SPC T(s)	PH T(s)	PH Gap%	PH No. It.	Open RPs
C1	2.50	1602.0	885.8	328.1	388.1	2.136	1.2	14.6
C2	2.61	3054.4	1897.6	446.7	710.1	2.544	1.4	14.6
C3	2.57	4563.2	2757.6	712.8	1092.8	2.442	1.4	15
C4	2.77	8325.6	5442.2	924.4	1959.0	3.452	1.8	16.6
C5	2.35	8790.2	5333.0	1605.0	1852.2	2.86	1.4	14.8
C6	2.62	10383.2	6157.6	2072.0	2153.6	2.44	1.2	14.2

instances is descending which means the efficiency of this algorithm improves by increasing the size of the problem. On average, 61% of the total runtime of PH-SLS is because of L-shaped and 22% and 17% because of PH and SPC, respectively. Average PH gap% is 2.65% implying that the PH result is reliable as this value is small, and, it can improve the efficiency of L-shaped embedded in the PH-SLS by helping L-shaped to start with a appropriate initial solution. This procedure cause L-shaped converge faster and generates better upper bounds. The PH results are attained by performing 1.4 iterations on average. Finally, the average number of open RPs is about 15 RPs and we do not see notable variations between test classes meaning that, for the considered geographical region, 15 open RPs seem enough.

4.5 Conclusion

In this study, we address the RP-network design problem under demand uncertainty. It is very common that the demand is not deterministic in the transportation network, hence we devise a framework to design an RP-network with uncertain or stochastic demands. Our problem assumptions and settings are exactly the same as those of our previous work in the previous chapter of this dissertation. We suggest a two-stage stochastic program to model the considered problem in a stochastic setting. The proposed recourse function satisfy the fixed and complete recourse assumptions.

We propose a new version of the L-shaped method, namely, SC-based cut L-shaped method working based on a new scenario categorization scheme which we introduce in this study. We show that this version of L-shaped method can outperform two well-known versions of it, single-cut and multi-cut, to solve the proposed problem. Also, we suggest a number of enhancements for improving the performance of the proposed L-shaped method. The most important one is using a new version of the progressive hedging algorithm which we devise for SMIP to generate a reliable initial solution for L-shaped. We show that this technique can improve the runtime, the number of iterations and the initial gap percentage of L-shaped. Other than this, we suggest adding mean value problem-based cut (MVP) to the L-shaped master problem and we show that it can improve the lower bound of the algorithm properly. Also, we suggest using the shortest path cuts to improve the initial lower bound of the algorithm and show that adding these cuts can improve the runtime of the algorithm. Adding disaggregated cuts and early termination the L-shaped master problem are two other enhancements we suggest to improve L-shaped performance.

In terms of computational results, we show that our overall algorithm, namely, PH-SLS, can improve the runtime and gap% of the single-cut L-shaped method by 40% and 37%, respectively. PH-SLS can solve all the test instances. The original versions of L-shaped cannot solve the majority of large-size test instances while the performance of PH-SLS gets better by increasing the size of the problem. We discuss that all the suggested improvements play important roles in improving the performance of the L-shaped, but the PH is the most important one. Using SC-based cuts improve the runtime of S-cut L-shaped by 11%. The impact of adding MVP cut on runtime is about 10%. Embedding PH and shortest path cuts can improve runtime by 14% and 3.5%, respectively while adding both of them concurrently can enhance the runtime by 25% which is impressive. On average, about 61% of the total runtime of PH-SLS is because of the L-shaped part of the algorithm, whereas, this value for PH and SPC is about 22% and 17%, respectively.

Chapter 5

Simulation-based Optimization for Operational Decisions in Truckload Transportation Network (P3)

In this chapter, we develop a simulation framework which is able to simulate transfer of TLs from their origins to destinations by assigning the drivers to TLs on a given network. This model should be able to simulate day-to-day operations in TL transportation as closely as possible. Using this simulation model, it is possible to compare different dispatching approaches. We can run the model assuming that the procedure of dispatching is PtP. Alternatively, we can consider both direct shipment and network shipment approaches concurrently as we discussed in the first part of the project. This provide us the means of evaluating the RP-network approach for TL transportation.

In our simulation, we generate the demands of commodities randomly. By considering other practical assumptions such as load pick-up and drop-off wait times, equipment breakdowns, etc., we make the simulated model more realistic. We conduct the simulation in a discrete events setting in which the events occur at particular instants of time and change the state of the system. The state of the system in our model can be the status of assigning drivers to truckloads.

Considering different assumptions and requirements about assigning the drivers to the TLs, we can devise different models. In this regard, we can embed an optimization model to suggest the best settings to have a good dispatching results. In our framework, we implement and test different networks and different dispatching approaches. Also, we introduce a new concept for dispatching the load, namely, collaborative dispatching approach. In collaborative dispatching, we assume that the drivers also have an access to the loads' data, i.e., they have access to the TLs and their origins, destinations, pick-up and drop-off times, etc. Hence, they provide their preference about the loads. By incorporating these type of requirements into the optimization model in addition to previous ones we build a collaborative dispatching model which will be more powerful in considering the drivers' point of views and preference and lowering the driver turnover rate. This is important because when the dispatcher assign the loads to the drivers without considering their preference, it is possible that, these preferences are disregarded by the dispatcher making the overall system inefficient and the drivers unsatisfied by their job. We try to address this issue by suggesting the mentioned collaborative approach.

Figure 5.1 presents a good overview of the current workflow and collaborative one concurrently. The elements in red (text boxes) are the contributions of this project. Currently, the dispatcher based on the requirements and characteristics of the loads and drivers assign the loads to drivers. In this project, we suggest a solution procedure considering the drivers' point of views about the loads as well. In this regards, we include the load ranked lists prepared by the drivers in the model. Moreover, the results of our model can be used to build a predictive model to predict the driver turnover rate in the TL industry. By doing so, the dispatcher can dynamically check the turnover rate and try to consider the assignments alleviating this issue.

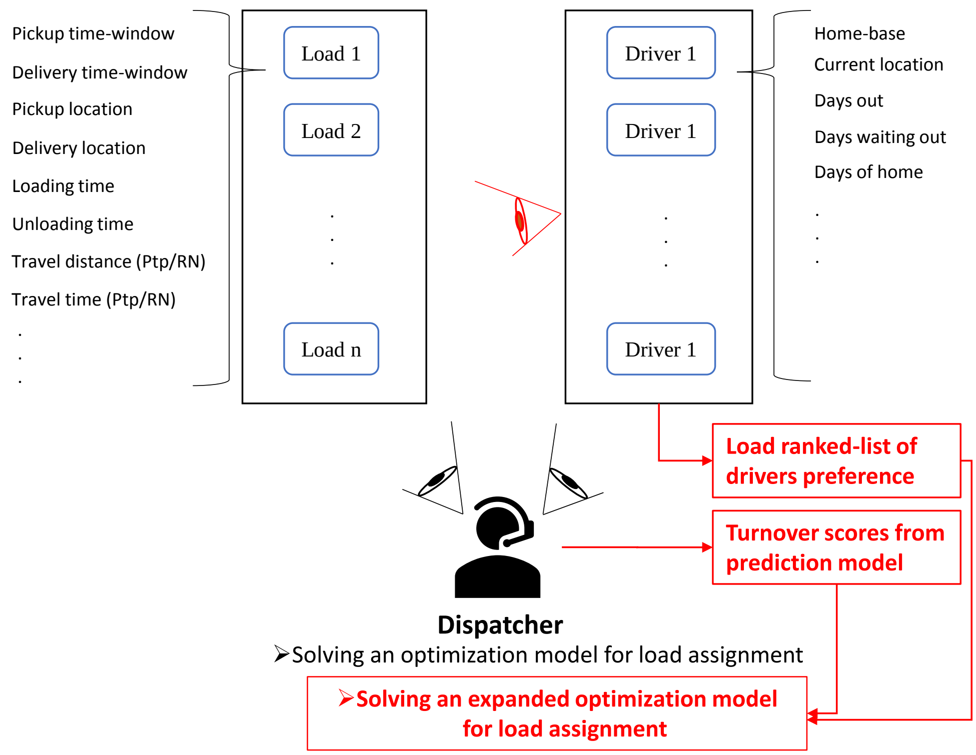

Figure 5.1: Overview of the data-driven collaborative simulation model

We will conduct a test framework to validate the proposed model and measure their efficiency in terms of the defined perfomance measure which we introduce later in the related section. These values can be used to predict the driver turn over rate. The simulation model provide us a practical environment that can predict the behaviour of the real industrial environment as much as possible. Using this system we can realize the industry's issues and bottlenecks and try to eliminate them by incorporating the improving suggestions and testing their impact and doing a extensive sensitivity analysis. Also, estimating the important practical features of the system is possible through the simulation models.

We explain the our simulation framework and its characteristics and details in Section 5.1. We present the design of our experiments in Section 5.2. Also, we present our experiments aspects and details and report and discuss the results in Section 5.3.

5.1 Simulation Method Development

We use discrete-event simulation method to model and test different networking and dispatching approaches and find the best ones. In discrete-event simulation we separate the time into sequential short periods of time and model the operations of the system in these time periods as a sequence of events.

Two major approaches to simulate the system are steady-state simulation and terminating simulation. In the first approach the performance of system is evaluated in the long-run and reaching to steady-state is a prominent requirement which should be regarded. In the second type, the performance of the system is examined in a limited period of time. Hence, a set of measures are defined. In our implementation, since we need to simulate the network for a set of loads delivered, we use the second approach. The runtime of our models starts by starting the simulation and finishes upon reaching to the delivery time of last load or entity. The important requirement regarding this type of simulation is well-defining the number of replications (or repeating the experiment, under the same condition). Setting a sufficient number for this, we can decrease the variability in experimental results and increase the accuracy of the estimate and the confidence level. We use box-plots (confidence intervals) to find the appropriate number of replications for our different models, as we explain later on in this chapter.

5.1.1 Networks and Their Properties

In this section, we investigate using two different types of networks, explained in Chapter 3. These two are Point-to-Point (PtP) or direct shipment network and Relay Point network (RPN) used to ship the loads. Figures 5.2 and 5.3 show this network in 2 and 3 dimensions, respectively, and Figures 5.4 and 5.5 show these for RPN.

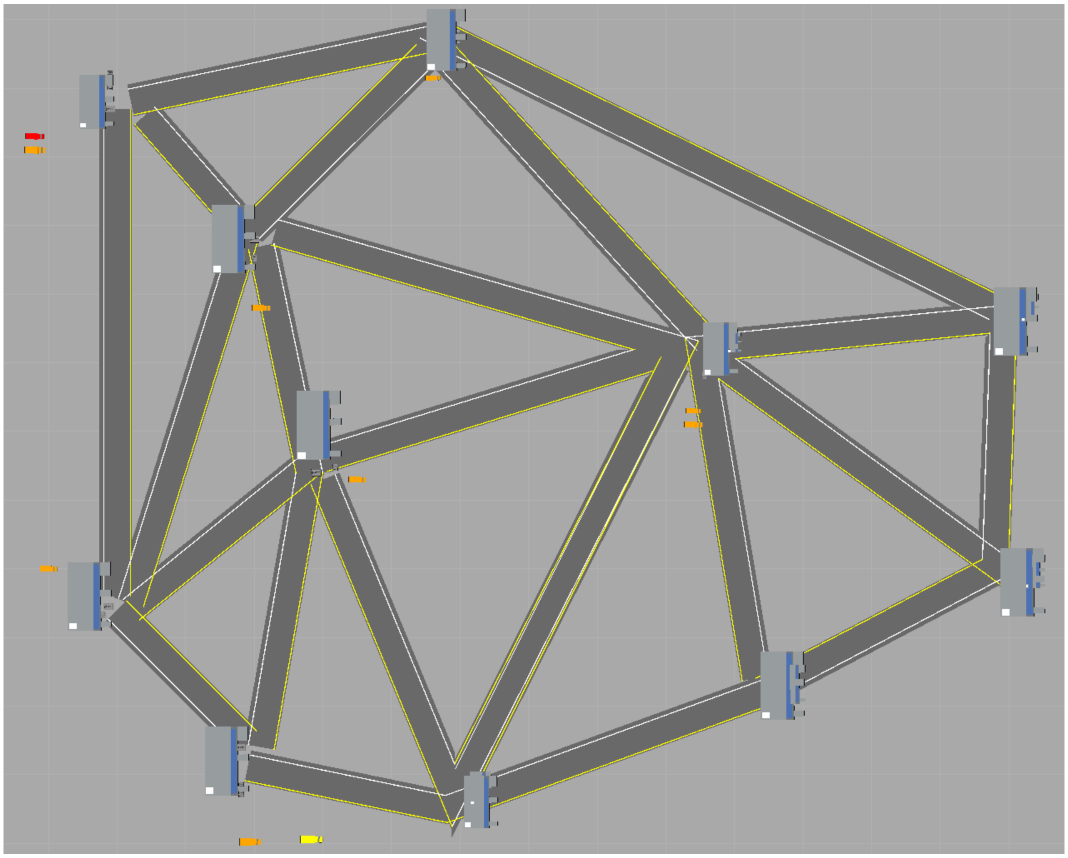

Figure 5.2: PtP_2D

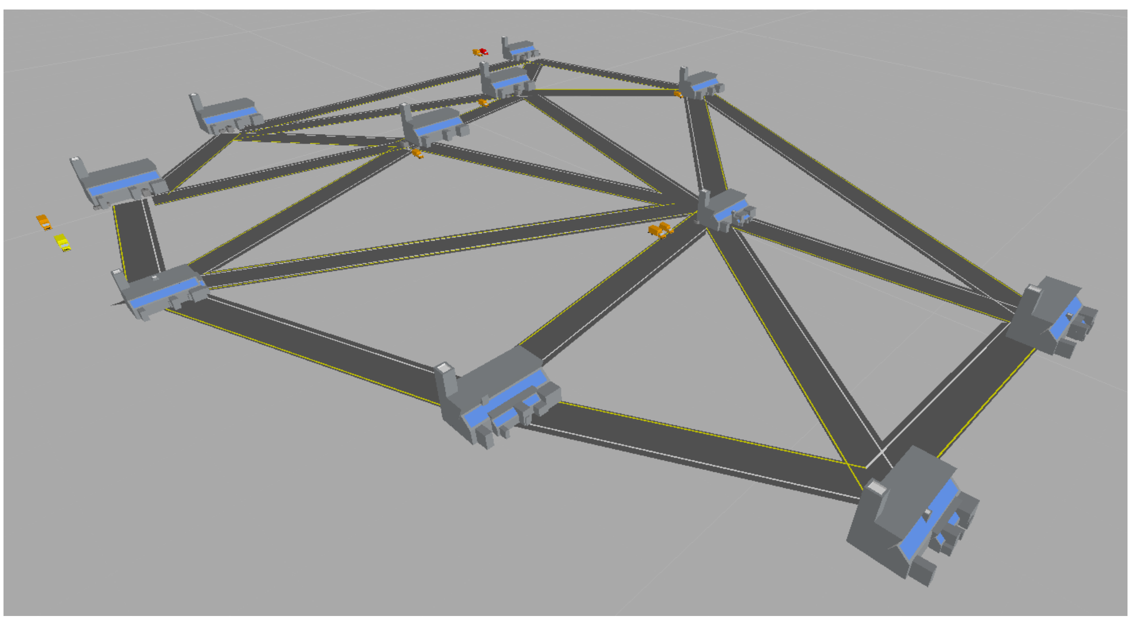

Figure 5.3: PtP_3D

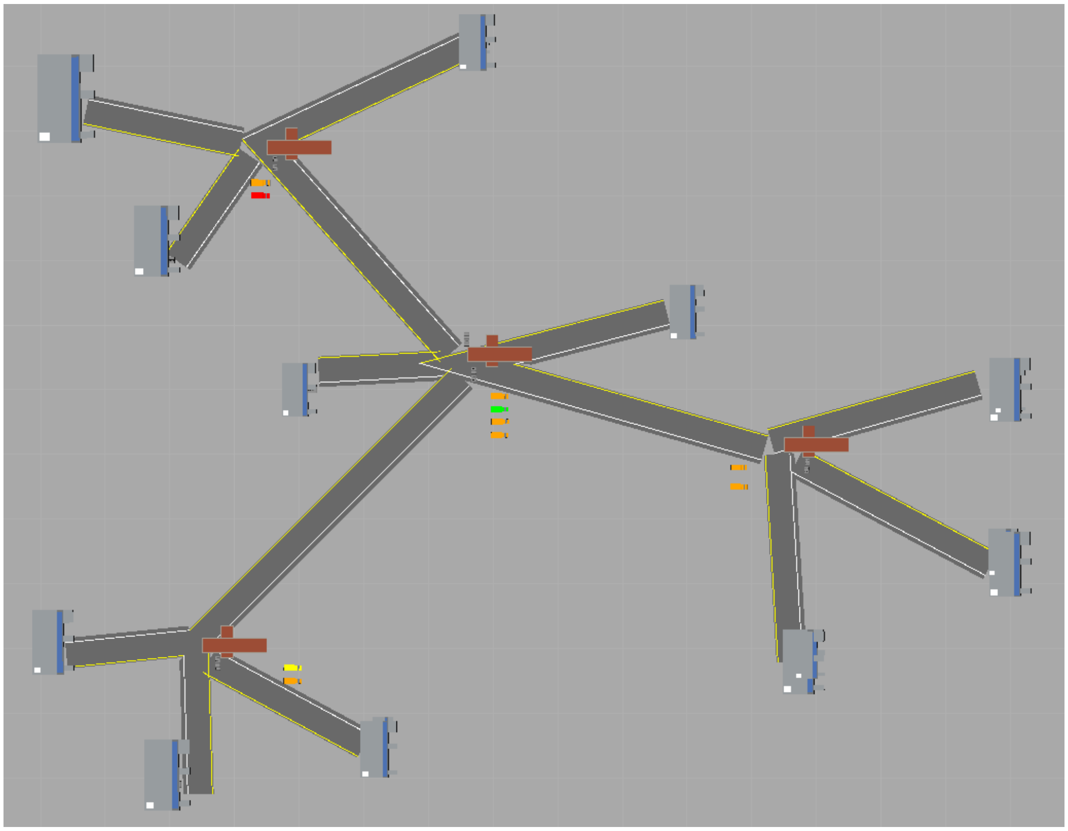

Figure 5.4: RPN_2D

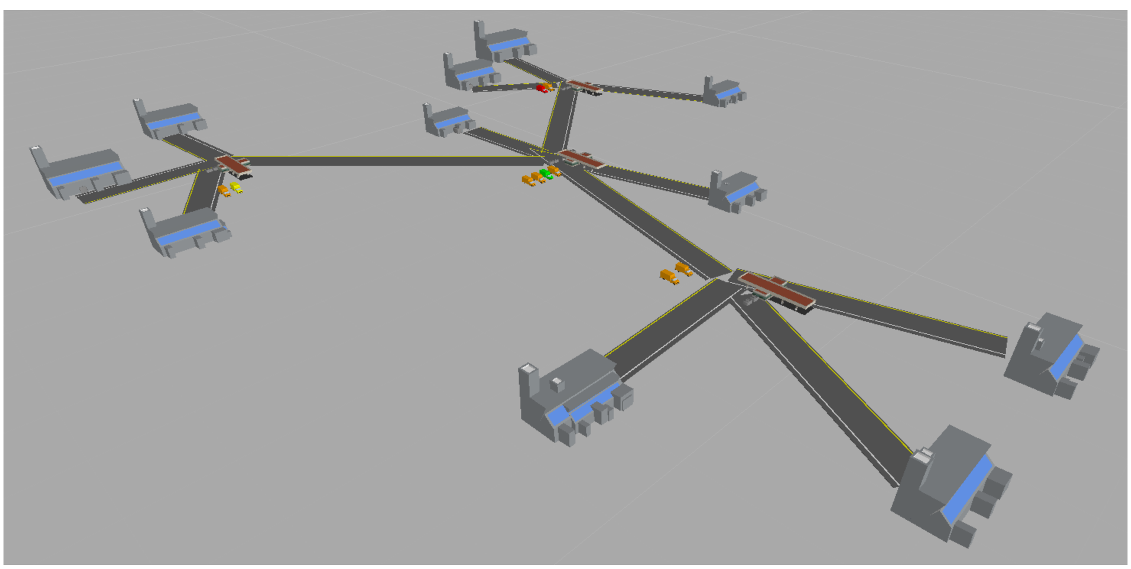

Figure 5.5: RPN_3D

In the PtP network each load is directly shipped from its origin to its destination. In other word, the direct shipment is the only method of shipment. Also, all the nodes are of type origin/destination node and can be origin or destination or both for the commodities defined. In the RP network two methods of shipment are available which are direct and network shipment. We have two types of nodes the origin/destination (O/D) nodes and RP nodes. In the Figures 5.4 and 5.5 the RP nodes are shown by a different symbol in red. It should be noted that in both networks the number and locations and all the properties of the O/D nodes are the same to be able to do a fair comparison. We just add a number of RPs to the RP network and change the roads according to the definition of the RP network. It should be noted that the RP-network which we consider in this chapter of this dissertation is a basic one which means the parallel shipment and multi-assignment assumptions are not considered in building it. The reasons for considering this type of RP-network is that adding these two assumptions builds a generalization of the RP-network such that this general version inherits all the performance characteristics of the basic version and usually perform better in practice, as we discussed before. On the other hand, adding these assumptions to the simulated model does not change the complexity of the model considerably and can be done easily. Hence, we prefer to perform our experiments on the basic version of this network to show the fundamental strangeness of this type of network.

We define the properties of the two models defined above. The process of building these networks are similar, but they have a few differences which we specify in the related section.

5.1.1.1 Entity Generation

The source nodes are responsible to generate the entities. The attributes of a source node 1 can be seen in Figure 5.6, as an example.

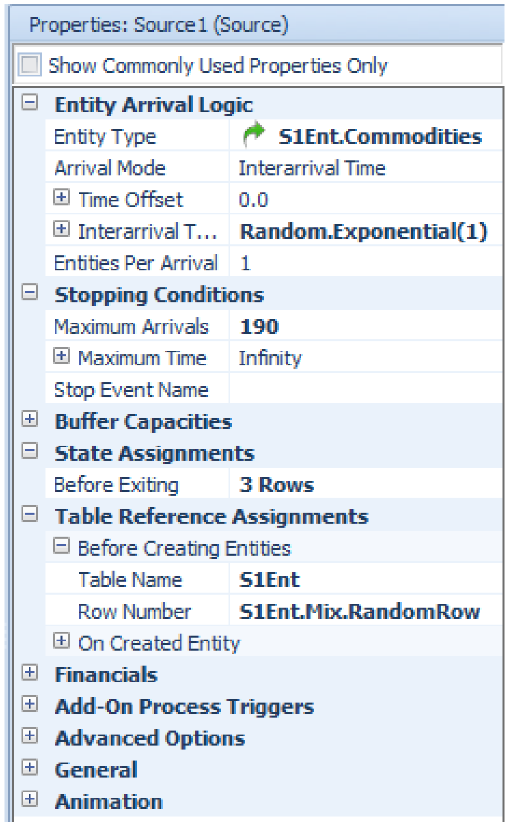

Figure 5.6: Source Node properties

We define "Data Table" to define various entities for the network sources. Figure 5.7 shows the data table for source 1. The entities are mixed equally in each source node. We add a column, namely, "Destination" to determine the destination of each entity type. Note that the table name and entities generating fashion, which is random-mix, are specified in "Table Reference Assignment" section.

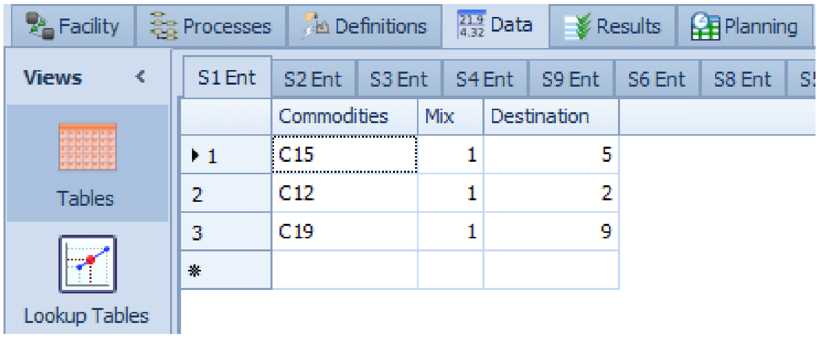

Figure 5.7: Source node data table

We define one State Assignment to specify the destination for each entity. To do so, we use the associated column presented in Figure 5.7. Figure 5.8 shows more detail of this implementation.

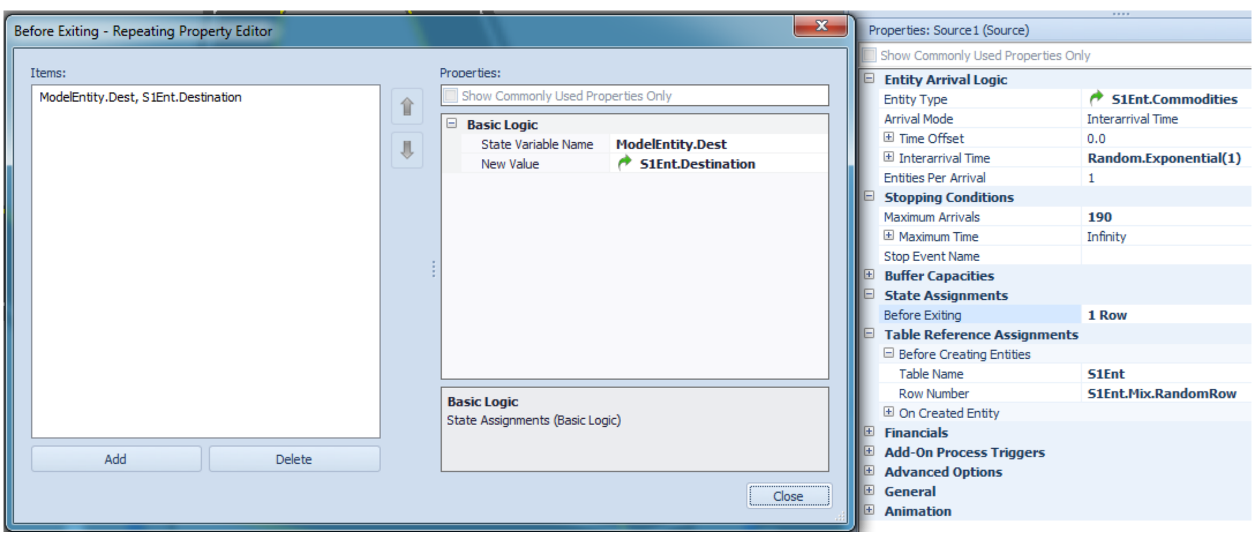

Figure 5.8: Source node state assignment

5.1.1.2 *Routing Strategy*

In order to conduct the entities to their specific destinations we follow the following procedure. First, we build an add-on process for each Output@Source node which assign each entity to another transfer node devised for each specific destination based on the destination-based state assignment described above. Figure 5.9 shows an example for this add-on process type for Source 1 in our model.

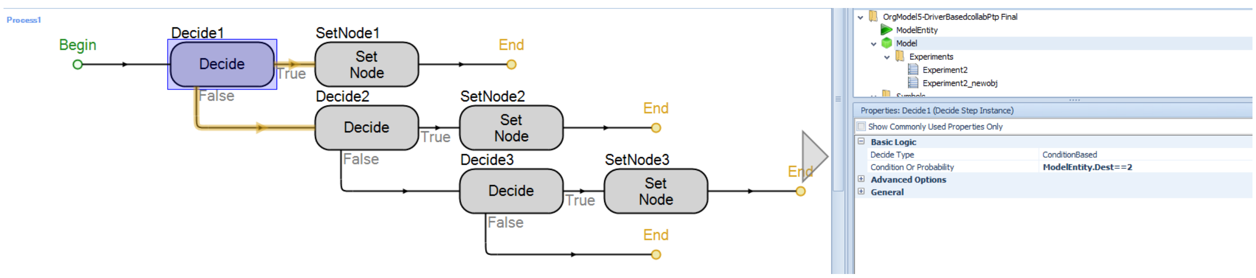

Figure 5.9: Routing add-on process

Afterward, the mentioned transfer node can be set such that it can handle the routing of each entity. To do so, we set the "Entity Destination Type" to "Specific" and enter the name of the destination in the "Node Name field",i.e., Input@Sink9.

In the PtP setting, the long-distance commodities usually use some intermediate nodes to reach to their destination. We devise a transfer node for each origin/destination node serving as intermediate node. The "Entity Destination Type" should be set to "Continue" for these type of nodes. For RPN, we do not need these kind of nodes.

5.1.1.3 *Using Vehicles to Pick-up and Deliver the Loads*

We use vehicles to transport the loads. Using vehicles in Simio needs several settings that should be considered. We define a number of vehicles to be used in the network. Figure 5.10 shows the settings for Vehicle 3 in our networks.

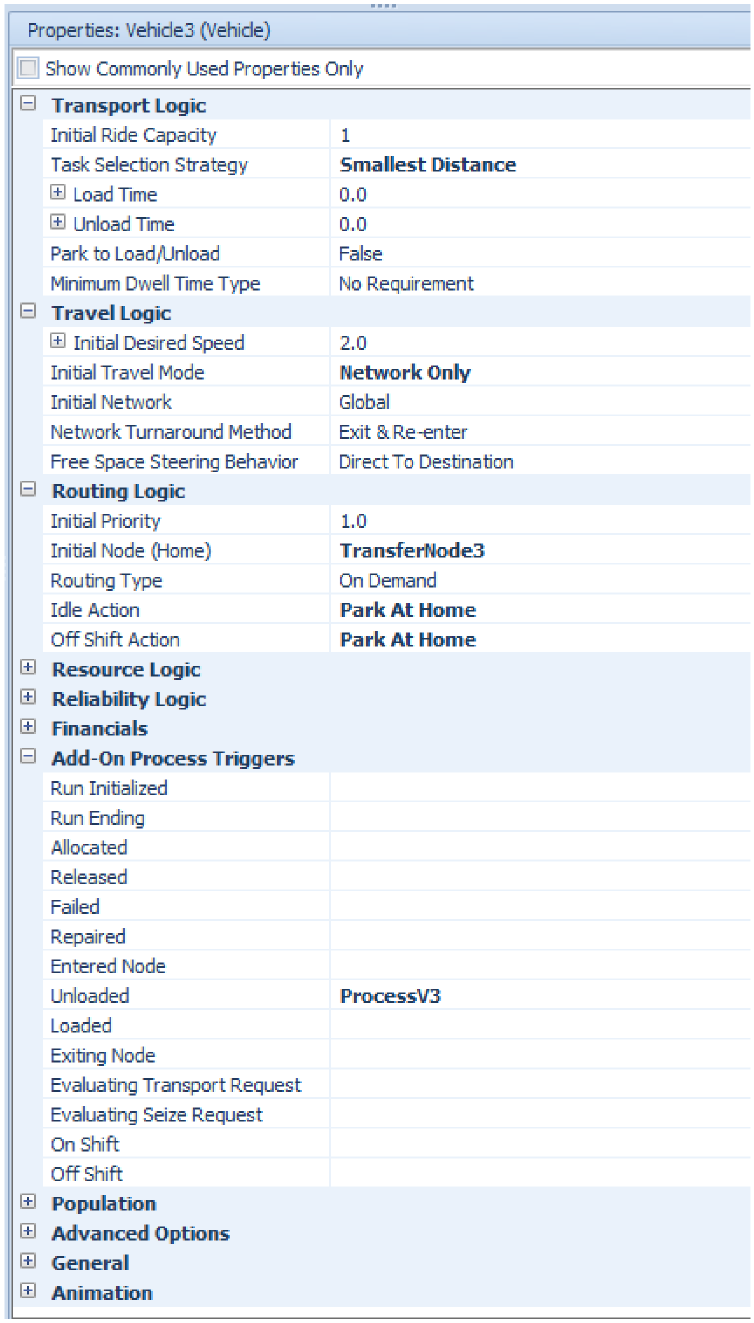

Figure 5.10: Vehicle properties

The first setting is changing the "Task Selection Strategy" which we discuss different settings for this field later in Section 5.1.2. We need to change "Initial Travel mode" to "Network Only" to force the vehicles use the available paths in the network. We should determine the vehicles home base through "Initial Node (Home)" and we set "Park At Home" for two fields "Idle Action" and "Off Shift Action" to let the vehicle park at home when they are not required during the simulation process.

One of the major assumptions of our models is the vehicles' or drivers' time-offs. To do so, we consider scheduling a time-off to each driver which has delivered a predetermined number of loads which means we force the drivers to come back home and take the time-off, but it is possible for them to deliver a few loads on their way to home. In order to consider this assumption, we devise the following procedure. First, we define one state variable for each vehicle, i.e."veh3load" for vehicle 3. This variable is considered as a counter to monitor the number of deliveries for the vehicle between to successive time-offs. We use this to build an add-on process for each vehicle showed in Figure 5.11 for Vehicle 3.

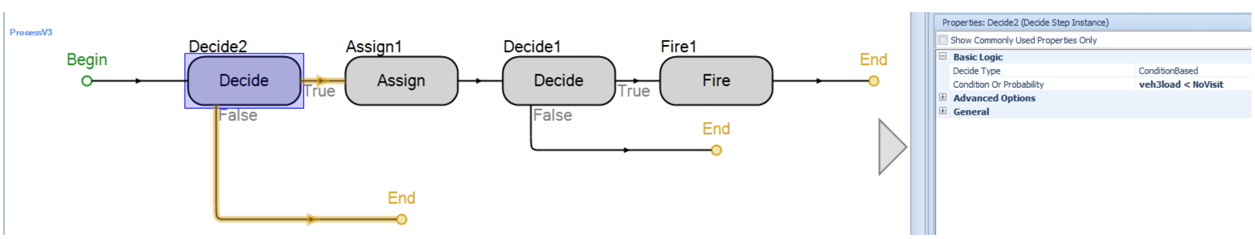

Figure 5.11: Vehicle add-on process

The last module of this add-on process is for firing an event, called Veh3, for Vehicle 3 to generate a time-off for this vehicle. To do so, we put an auxiliary source and sink close by the vehicle home base, and after firing the event, we generate one auxiliary entity which should be delivered to the mentioned destination by the vehicle. Figure 5.12 shows the required settings for the mentioned source/sink in which CV3 is the mentioned auxiliary entity. Also, the "TimeOff" is an Expression Property which we define to control the length of time-off for vehicles. We use this property as the input for field "Transfer-in Time" of the mentioned auxiliary sink node. In this regard we define another Expression Property, namely, "NoVisit" to control the number of deliveries before scheduling a time-off for any vehicle.

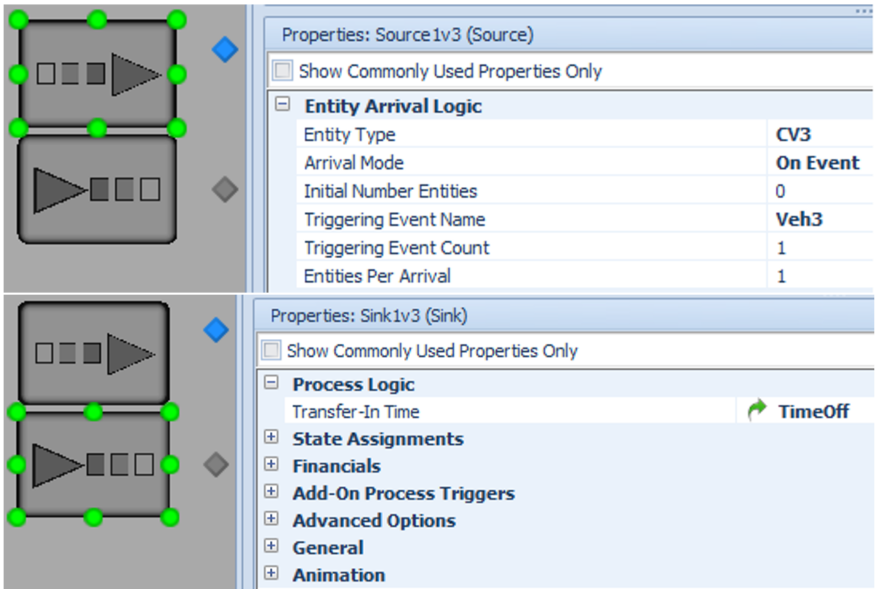

Figure 5.12: Vehicle time-off settings

5.1.2 Dispatching Strategies

In this section we describe two different dispatching approaches. Implementing these two approaches for the two models described above needs different procedures. Hence, in this section, we describe four different implementations.

Dispatcher-based dispatching approach(DbD): It is usual that the dispatcher decides about the assignment loads to drivers without considering the drivers' preferences in term of selecting the loads. In this study, we call this approach "dispatcher-based" dispatching approach. The procedures of employing this approach for the two models are described. The main assumption in this study about the preference of dispatchers in reserving a vehicle for a load,at the time, is the one having the shortest distance from the location of the load.

Collaborative dispatching approach (CD): In this study, we introduce another approach to assign the loads to drivers. In this approach after assigning the loads to drivers they will be able to select the loads based on their preference. Similar to dispatchers, the main assumption of the models in this section is that the drivers prefer to pick a load having the shortest distance from their location among the assigned loads.

5.1.2.1 Dispatcher-based Dispatching Approach for PtP Model

To simulate this approach, we first need to create a driver list. Afterward, any transfer node devised to conduct an entity should be set as Figure 5.13 showing this setting for TransferNode2 used for entity C19.

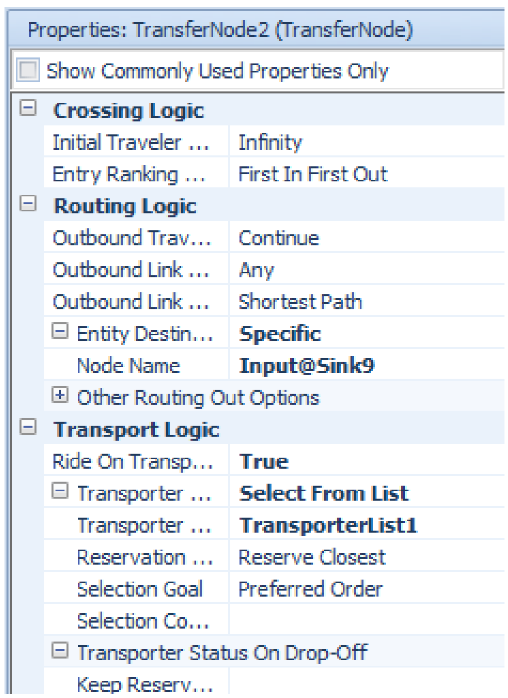

Figure 5.13: Load dispatching settings

The major settings in this context are, the "Ride On Transporter" should be set to "True", the "Transporter Type" to "Select From List" and "Transporter List Name" to the name of transporter list which is "TransporterList1". The important point is that the "Reservation Method" should be set to "Reserve Closest". Having this setting always let any available load reserve the closest vehicle and force it to pick-up and deliver it. In other words, the closest vehicle is reserved to serve the mentioned load. This behaviour simulate the dispatcher-based dispatching approach. It should be noted that the "Task Selection Strategy" for vehicle should be set to "First In Queue" which means each vehicle will serve the first load waiting in the queue.

5.1.2.2 Dispatcher-based Dispatching Approach for RPN Model

Similar to section above, we need to use transporter list property, but not only one list. We need to create several lists. We define two types of transporter lists, one for local transportation and one for lane one. The local lists are used by origin/destination node and RPs, and lane type lists are used by neighbor RPs. Figure 5.14 shows the defined lists for our model.

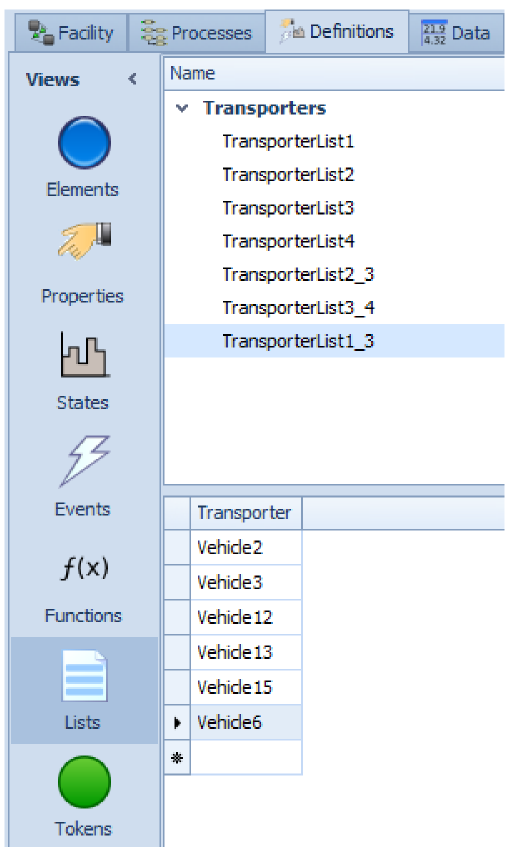

Figure 5.14: Local and Lane transporter lists for dispatcher-based approach in RP-network

In this figure, the "TransporterList1" is of type local and is used by RP 1 and all covered nodes, and "TransporterList1_3" is of type lane and is used by RPs 1 and 3. The other settings in this model are as dispatcher-based dispatching approach for PtP model settings described above.

5.1.2.3 Collaborating Dispatching Approach for PtP Model

To simulate this behaviour in the PtP model, we define several transporter lists which all of them contain all the vehicles, but in different orders. We define a list of transporters for each node in the network. The order of a list assigned to the node is based on the distance of vehicles' home base from that node in ascending order. Figure 5.15 shows all the lists defined for this model and the order of TransporterList1 assigned to Node 1 which we first put the vehicles of node 1 and then the vehicles of node 3.

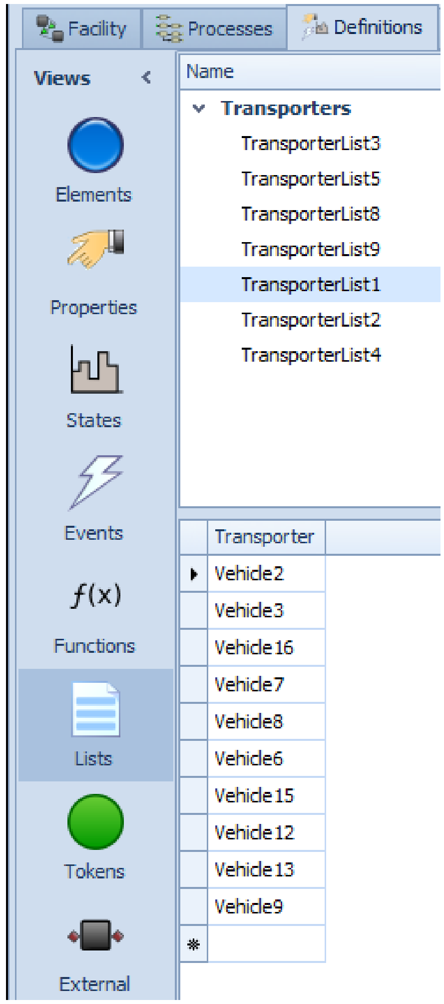

Figure 5.15: Transporter lists for collaborating dispatching approach in PtP model

The two other important settings considered in this model to simulate collaborative approach are, firstly, setting "Task Selection Strategy" for all the vehicles in this model to "Smallest Distance" to let them select the closes assigned load by dispatcher, and setting the "Reservation Method" for selecting vehicle in Transporter Logic of the origins to "Reserve Best". The latter setting force the dispatcher to follow the assigned transporter list order to find an unreserved vehicle and assign the on-hand load to it. Hence, the order/s of vehicles in the transporter list/s is/are very important and should be set accurately.

5.1.2.4 *Collaborating Dispatching Approach for RPN Model*

The settings of this model are very similar to previous one. The only difference is in the type of transporter lists. In this model, the type of the transporter list are similar to the model RPN dispatcher-based approach, but we double the lane related lists to consider different order for two neighbor RPs based on the preference. Figure 5.16 shows the transportation list for this model. As it can be seen in the example shown in the figure, for two RPs 2 and 3 we define two lists, one for when the loads start form RP 2 and one for vice versa.

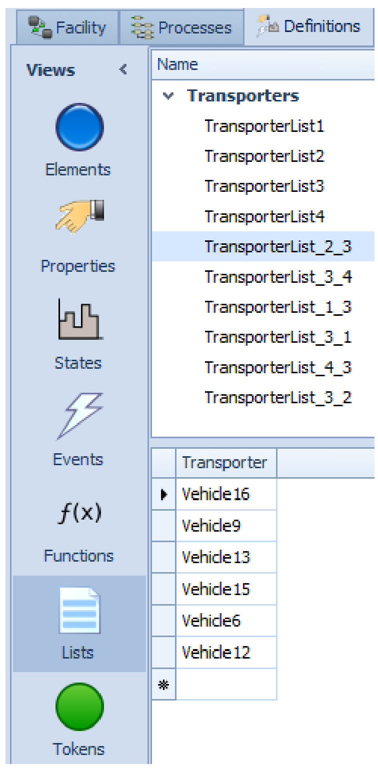

Figure 5.16: Transporter lists for collaborating dispatching approach in RND model

5.1.3 Timer Setting

In this study, the runtime of the different models are different. In other words, the simulation starts by pressing the run button and ends by delivering the last entity. To do so, we set the "Ending Type" field to "Unspecified(Infinite)". In this situation, to have exact run length for the models we devise the following procedure. First, we define a timer element, namely, Timer 1. Afterward, we define three state variables, "RunEndTime", "RunLength" and "TotalDemand" to build an add-on process to capture the run length. Figure 5.17 shows the mentioned process.

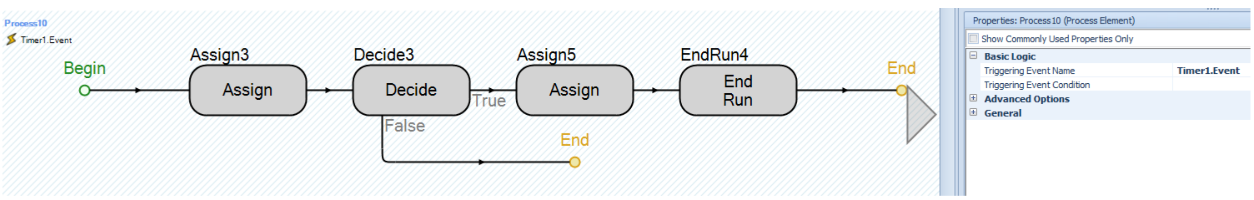

Figure 5.17: Timer add-on process

5.2 Design of Experiments and Performance Measures

In this section, we describe our procedure to design a number of scenarios to test the performance of different models and select the best setting with respect to the defined performance measures defined in this section. In this regard we define two variables as follows:

NoVisit	the number of deliveries before scheduling a time-off for each driver
TimeOff	the length of each time-off (m).

Changing these variables makes different values for the considered measures and objectives. Hence, we need an optimization procedure to find the best value for these two variables. One way of doing optimization is performing simulation-based optimization which can be implemented by Simio. To do so, we select two sets of values for two input values and create a number of scenarios. We run the simulation models for each scenario. The model having the best average of objective value is selected as the best model. Also, the scenario of the best resulted values among all the simulation runs is selected as the best case and the variables values are suggested as the best solution.

To test the performance of the models, we define three different measures. These three measure can take both the drivers and company points of view into account. These three measures are as follows:

AvgGetHome	average number of scheduled time-off for the vehicles
AvgOnShift%	average time percentage each vehicle is needed
AvgTimeInSystem	average time each entity (TL) spends in system.

The first two measure are used to calculate the get-home-rate for the drivers which is an important element to predict the driver turnover rate. The first measure plays an important role in regularizing the drivers schedule in addition to the expanding length of time-offs for the drivers. This point is also important in term of decreasing the driver turnover rate. The third measure is important for transportation companies, since it is directly connected to delivery date and operational costs.

Also, in our implementation, the runtime of different models are different because we consider the time of delivering the last entity of the model as the ending time. To capture this time period we define another measure as follows:

RunLength	runtime of the model (time between TL generation time to its delivery time).

We also suggest a number of measures built based on the introduced measures and a performance measure including all of them, as a single objective for the problem, later in this section.

5.3 Computational Results

To evaluate the performance of the models described above, we design a number of experiments. In these experiments, we try to evaluate the main elements of the proposed models and compare the performance of the models. To do so, we first explain the commodity generation procedure in which we suggest to create a number of demands to be satisfied. Afterward, we discuss the numerical results obtained in detail. Finally, we propose a procedure to find the best solution or setting resulting in the best outputs.

All of the experiments are conducted using Simio 8 on machines with an Intel Core i7-4790 CPU at 3.6 GHz, 32 GB RAM running 64-bit OS.

5.3.1 Commodity Generation

In this section, we demonstrate the entity generation procedure which we devised to generate commodities for all the models. In our experiments, we generate 23 different entity types. The number of entities generated of each type are as Table 5.1. Totally, 2000 TLs are delivered in the experiments.

All the source nodes use Exponential distribution with the mean of 6 minutes to generate the specified number of entities presented in Table 5.1. Note that for the sources generating more than one entity types, the share of the types are equal.

5.3.2 Replications Requirements

As we mentioned above, in the terminating-simulation the number of replications plays an important role in accuracy of the results. In this study, we follow a procedure to set the number of replications appropriately. In this approach, we start by a small number of replications and plot the associated box-plot. Afterward, we increase the number of replications and again check the box-plots. We continue this process to reach to a reasonable box-plot and report the associated number of replications as the final number of replications. We stop increasing the number of replications when the three associated confidence intervals stop overlapping. In this state we can be sure that the final results are reliable and accurate. Figure 5.18 shows this process for one of our models, as an example. We tested six different numbers, 10, 20, 30, 40, 50, and 60 as the Scenario 1 to 6 for one of our measure, namely, AveTimeInSystem.

Table 5.1: Entities Specifications

Entity Name	Destination	Origin	No. of Entities Generated
C12 C15 C19	Sink2 Sink5 Sink9	Source1	190
C23 C28 C212	Sink3 Sink8 Sink12	Source2	210
C34 C38 C312	Sink4 Sink8 Sink12	Source3	200
C48	Sink8	Source4	210
C54 C56	Sink4 Sink6	Source5	220
C65	Sink5	Source6	170
C85 C89 C810 C811	Sink5 Sink9 Sink10 Sink11	Source8	200
C95 C96	Sink5 Sink6	Source9	220
C103	Sink3	Source10	190
C111	Sink1	Source11	190
C1210 C123	Sink10 Sink3	Source12	200

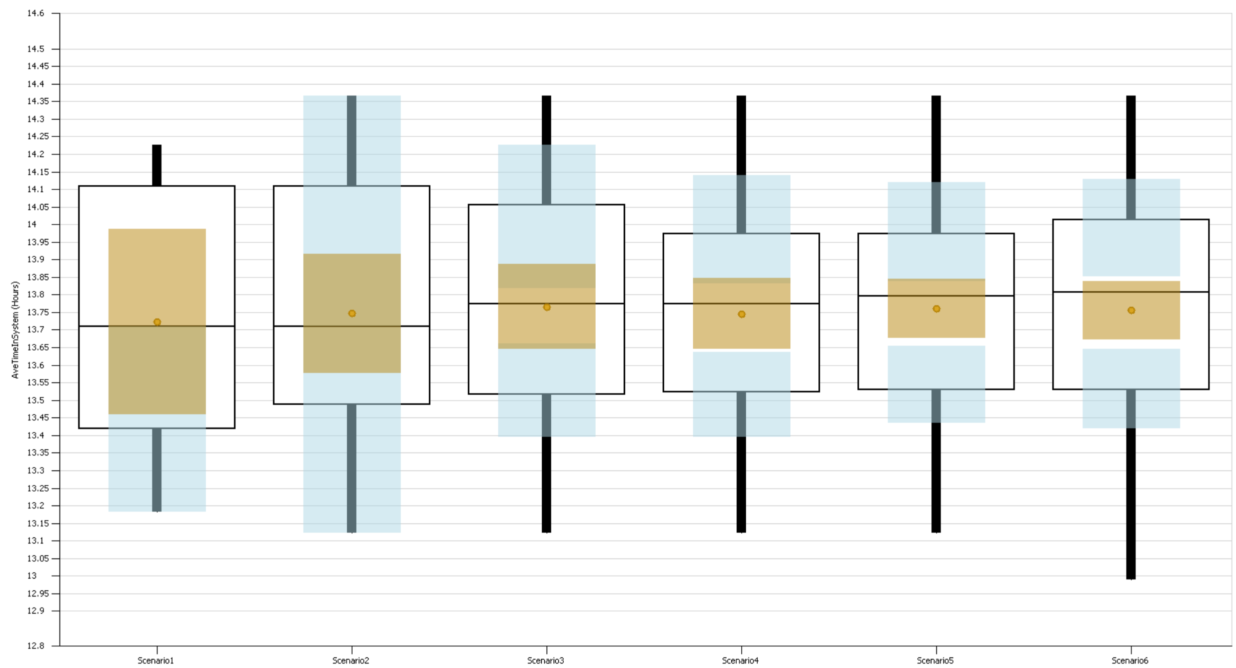

Figure 5.18: Box-plots to set the number of required replications

The only chart in which three confidence intervals "Upper Percentile Confidence Interval","Mean Confidence Interval" and "Lower Percentile Confidence Interval" (the three boxes showed in the figure, from up to down) are not overlapping is the Scenario 6. Hence, 60 is a good number for this case. Since we see this behaviour for all our models on all the considered responses, we report 60 as the final number of required replications and use it for our all experiments. It should be noted that the confidence level for all of the implementations is set 95%.

5.3.3 Numerical Results and Discussion

In this section we present the results of implementing different models. Based on the measures above, we create a number of higher level measures reporting the performance of the models in a better way. These new measure are as follows.

$TotalTimeOff(m)$: reports the average of total time in (minutes) which each vehicle is not working.

$$TotalTimeOff(m) = AvgGetHome \times TimeOff(m) \\ + AvgOffShift\% \times RunLength(m) \quad (5.1)$$

where, $AvgOffShift\% = ((100 - AvgOnShift\%)/100) \times RunLength(m)$. This equation calculate the total time-off for a driver consisting two major parts: the total scheduled time-offs, which is the total time of the scheduled time-offs, and the total length of another type of time-off which is assigned to the drivers whenever they are not needed in the system.

$TotalTimeOff\%$: reports the average percentage of time in which each vehicle is not working.

$$TotalTimeOff\% = \frac{TotalTimeOff(m)}{RunLength(m)} \times 100 \quad (5.2)$$

$AvgTimeInSystem\%$: reports the average percentage of time in which each TL is in the system.

$$AvgTimeInSystem\% = \frac{AvgTimeInSystem(m)}{RunLength(m)} \times 100. \quad (5.3)$$

These three measures are important since using them, we can evaluate the performance of different models properly.

In order to create a number of scenario we select two appropriate ranges for our two input variables. These ranges are determined in accordance with other parameters ranges of the models. We select the values for "NoVisit" from set $\{2, 5, 10, 20\}$ and for "TimeOff" from $\{5, 10, 15, 20\}$. using these values, we create sixteen different scenarios for which we simulate the models and report the results. Tables 5.2 to 5.5 shows the results of PtP with DbD, PtP with CD, RPN with DbD and RPN with CD model, respectively, for different scenarios. We first discuss the behaviour of the network types, and then we discuss the performance of different dispatching approaches.

Table 5.2: Results of PtP with DbD model

Scenario	NoVisit	TimeOff(m)	TotalTimeOff%	AveTimeInSystem%	RunLength(h)
1	2	5	0.38	38.13	88.57
2	5	5	0.33	37.99	88.39
3	10	5	0.36	37.94	88.30
4	20	5	0.23	37.91	88.18
5	2	10	0.62	38.18	88.73
6	5	10	0.54	38.03	88.58
7	10	10	0.53	37.93	88.40
8	20	10	0.33	37.94	88.26
9	2	15	0.87	38.27	88.92
10	5	15	0.75	38.10	88.77
11	10	15	0.67	37.96	88.51
12	20	15	0.47	37.96	88.24
13	2	20	1.11	38.34	89.13
14	5	20	0.98	38.12	88.96
15	10	20	0.84	37.97	88.63
16	20	20	0.62	37.94	88.40
Average	9.25	12.50	0.60	38.05	88.56

Table 5.3: Results of PtP with CD model

Scenario	NoVisit	TimeOff(m)	TotalTimeOff%	AveTimeInSystem%	RunLength(h)
1	2	5	2.89	23.31	59.00
2	5	5	1.95	23.45	58.02
3	10	5	1.52	23.45	57.60
4	20	5	1.18	23.42	57.29
5	2	10	5.09	23.37	60.43
6	5	10	3.41	23.58	58.89
7	10	10	2.46	23.55	58.18
8	20	10	1.76	23.49	57.65
9	2	15	7.12	23.43	61.80
10	5	15	4.67	23.67	59.74
11	10	15	3.32	23.64	58.73
12	20	15	2.38	23.54	58.04
13	2	20	8.96	23.44	63.16
14	5	20	5.91	23.76	60.58
15	10	20	4.22	23.72	59.30
16	20	20	2.94	23.60	58.40
Average	9.25	12.50	3.74	23.53	59.18

Table 5.4: Results of RPN with DbD model

Scenario	NoVisit	TimeOff(m)	TotalTimeOff%	AveTimeInSystem%	RunLength(h)
1	2	5	10.04	54.66	123.00
2	5	5	9.99	54.60	123.05
3	10	5	9.92	54.57	122.96
4	20	5	9.90	54.53	122.86
5	2	10	10.49	54.78	123.69
6	5	10	10.39	54.70	123.59
7	10	10	10.27	54.64	123.42
8	20	10	10.22	54.54	123.33
9	2	15	10.94	54.92	124.38
10	5	15	10.76	54.79	124.14
11	10	15	10.61	54.69	123.85
12	20	15	10.52	54.57	123.74
13	2	20	11.38	55.01	125.01
14	5	20	11.13	54.88	124.64
15	10	20	10.91	54.77	124.32
16	20	20	10.87	54.58	124.26
Average	9.25	12.50	10.52	54.70	123.76

Table 5.5: Results of RPN with CD model

Scenario	NoVisit	TimeOff(m)	TotalTimeOff%	AveTimeInSystem%	RunLength(h)
1	2	5	18.29	33.29	95.00
2	5	5	10.15	32.65	85.14
3	10	5	6.52	32.30	81.75
4	20	5	4.14	32.06	79.69
5	2	10	29.84	34.26	110.12
6	5	10	17.45	33.55	92.38
7	10	10	11.06	32.94	85.75
8	20	10	6.83	32.41	81.96
9	2	15	38.29	34.96	124.90
10	5	15	23.66	34.14	99.94
11	10	15	15.29	33.45	89.94
12	20	15	9.40	32.79	84.03
13	2	20	44.97	35.31	139.80
14	5	20	28.84	34.76	107.00
15	10	20	19.02	33.91	94.08
16	20	20	11.74	33.15	86.32
Average	9.25	12.50	18.47	33.50	96.11

5.3.3.1 Collaborative vs. Dispatcher-based Dispatching Approach

In this part we discuss the results of two introduced dispatching approaches on two different models. First, we compare the results of PtP approach. The average runtime of CD is about 33% less than DbD which is impressive. Other than this CD can increase the total per driver time-off percentage by 522%. Also, CD can decrease the average time in system for each entity by 38%. Hence, our discussion showed that the CD approach is strictly better than DbD since it can decrease the overall delivery time while it increasing the overall time-off rate for the vehicles.

The results of testing two approaches on the RPN system reveals that CD can decrease the average run length and the average time in system for each entity by 22% and 39%, respectively. Also, it can increase the total per driver time-off percentage 76%. Similar to PtP model, using CD approach is strictly better than DbD.

The reasons why the performance measure are significantly better in the CD approach can be having less empty mileages and better assignment of loads to drivers. Considering these two discussions, we continue our experiments by incorporating the CD approach for our further implementations.

5.3.3.2 Point-to-Point vs. RP-Network

Having the previous section evaluations, we compare the results of RPN and PtP models using CD approach. The run length and the average time in system for each entity of PtP approach are about 38% and 30% less than RPN approach, but the total per driver time-off percentage of PtP is about 20% of the RPN model which means using RPN, we can increase this measure by four times which is impressive.

Regarding the results above, we can verify our discussion about the impact of using RPN networks to decrease the driver turnover rate in the TL industry. But, as we discuss the run length of RPN is higher in the RPN networks which can have a couple of reasons such as circuity issue, increasing the number of layovers and of course having a higher time-off for drivers.

To improve the runtime in RPN network, we suggest increasing the number of vehicles. We test the impact of this increase as we show in Table 5.6 the results of increasing the number of vehicles at each RP by one vehicle (total four vehicles). As it can be seen the average run length for RPN in this setting is almost similar to PtP model while we maintain the average per vehicle time-off percentage same as before.

Table 5.6: Result of RPN with CD having additional vehicles

Scenario	NoVisit	TimeOff(m)	TotalTimeOff%	AveTimeInSystem%	RunLength(h)
1	2	5	18.67	29.55	67.41
2	5	5	10.64	28.37	60.75
3	10	5	6.89	27.76	58.30
4	20	5	4.49	27.31	56.89
5	2	10	30.35	31.19	78.51
6	5	10	18.05	29.65	66.12
7	10	10	11.47	28.63	61.21
8	20	10	7.17	27.81	58.45
9	2	15	38.84	32.36	89.12
10	5	15	24.13	30.72	71.26
11	10	15	15.58	29.42	64.09
12	20	15	9.68	28.37	59.99
13	2	20	45.35	33.15	99.66
14	5	20	29.38	31.53	76.59
15	10	20	19.40	30.13	67.16
16	20	20	12.07	28.83	61.57
Average	9.25	12.50	18.89	29.67	65.92

This finding is interesting since can help us to conclude that using the RPN and increasing the number of vehicles properly, we can manage both the get-home times and promise date to the customers and satisfy both drivers and transportation company.

5.3.4 Simulation-based Optimization for the Selected Model

Determining the best model and setting, we are able to do optimization to select the best scenario among all the defined scenarios. To do so we use the OptQuest module of Simio to do optimization. The setting of OptQuest is as Figure 5.19.

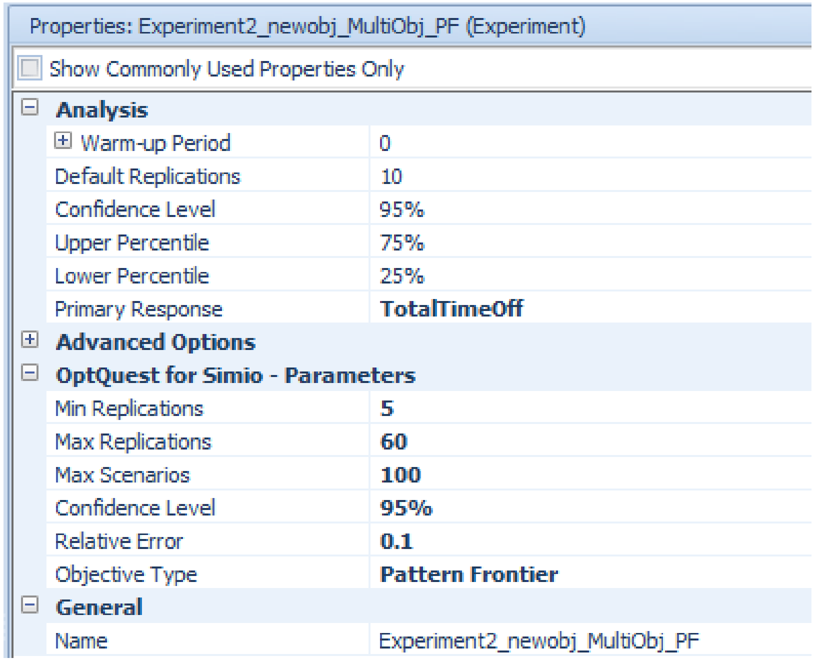

Figure 5.19: OptQuest setting for Pattern Frontier multi-objective simulation-optimization

We use OptQuest add-in to generate a number of good scenarios to find the best values for two input variables. We set the range [2, 20] and [5, 20] with increment value equal to 1 for “NoVisit” and “TimeOff(m)”, respectively. Since we have two objectives, “TotalTimeOff(h)” and “AveTimeInSystem(h)”, with different behaviour, we use the “Pattern Frontier” objective type to do multi-objective optimization. We set 100 as the maximum number of required scenarios among all possible scenarios. Evaluating 100 scenarios, OptQuest selects 37 scenarios as the set of good and reliable scenarios. Table 5.7 shows these 37 scenarios.

In the last column of this table, we introduce and tested a performance measure to compare different scenarios. Equation (5.4) shows this measure formula.

$$PM = \frac{TotalTimeOff(m)}{AvgTimeInSystem(m) \times RunLength(m)^2} \times 10^8 \tag{5.4}$$

As it can be seen, it takes a relatively high value for the suggested scenarios by Simio. Hence, it can be used as an objective function which its maximum value can be considered as the

Table 5.7: Selected scenario set resulted by OptQuest

Scenario	NoVisit	TimeOff(m)	TotalTimeOff%	AveTimeInSystem%	RunLength(h)	PM
2	2	5	0.18	0.29	66.93	3.95
3	20	20	0.12	0.28	61.35	3.15
4	7	9	0.13	0.29	62.11	3.34
5	16	16	0.12	0.28	61.25	3.14
6	2	19	0.44	0.32	96.72	4.05
7	20	7	0.05	0.27	57.31	1.73
10	2	12	0.34	0.31	82.38	4.48
11	19	12	0.08	0.27	59.16	2.42
12	7	16	0.21	0.30	68.10	4.16
13	9	5	0.07	0.27	58.37	2.16
14	15	12	0.10	0.28	60.08	2.78
15	16	5	0.05	0.27	57.17	1.62
16	2	15	0.39	0.32	88.65	4.31
17	2	20	0.45	0.32	98.94	3.95
18	2	7	0.24	0.30	71.44	4.36
19	4	18	0.31	0.31	77.56	4.55
20	4	15	0.27	0.31	74.13	4.50
21	4	7	0.16	0.29	64.13	3.71
22	4	6	0.14	0.28	62.90	3.48
26	6	13	0.19	0.30	67.21	4.00
28	3	18	0.36	0.32	84.05	4.45
29	2	14	0.37	0.32	87.05	4.25
31	2	17	0.42	0.32	92.85	4.19
42	5	19	0.28	0.31	74.88	4.56
43	2	16	0.40	0.32	91.24	4.13
46	4	17	0.29	0.31	76.19	4.58
47	3	19	0.37	0.32	84.84	4.46
48	3	15	0.32	0.31	79.15	4.56
49	3	17	0.34	0.32	82.51	4.39
51	2	6	0.21	0.29	69.46	4.19
61	5	13	0.22	0.30	69.35	4.18
70	9	6	0.08	0.28	59.25	2.38
72	5	15	0.24	0.30	70.79	4.44
74	3	6	0.17	0.29	64.93	3.84
75	13	11	0.10	0.28	60.50	2.78
81	2	11	0.32	0.32	80.69	4.37
84	4	8	0.17	0.29	65.57	3.91
94	5	16	0.25	0.31	72.50	4.33
97	4	9	0.19	0.29	66.78	4.10
99	4	13	0.25	0.30	71.58	4.43

best objective for the problem and its associated inputs as the best solution.

Considering this new objective function, we conduct another simulation-based optimization procedure using both OptQuest and KN (the Simio packages for simulation-based optimization) to select an smaller set of solution can be considered as the best solution set. To do so we consider the PM as the only objective function of the problem. Also, we set a limit for "RunLength" less than or equal to 70(h), because we are interested in finding the best solution (with respect to the measures defined) having a reasonable runtime as well. In this regard, we use OptQuest to determine a set of good scenarios. Afterward, we use KN to find the best solution set. KN is a sequential procedure in Simio to select the best scenario set from a bigger set of candidate scenarios. The setting for KN we used is as Figure 5.20. The indifference zone is selected equal to 0.1 meaning all the best solution with the distance of 0.1 with respect to PM value are presented as the best solution set. Table 5.8 shows the results of this procedure.

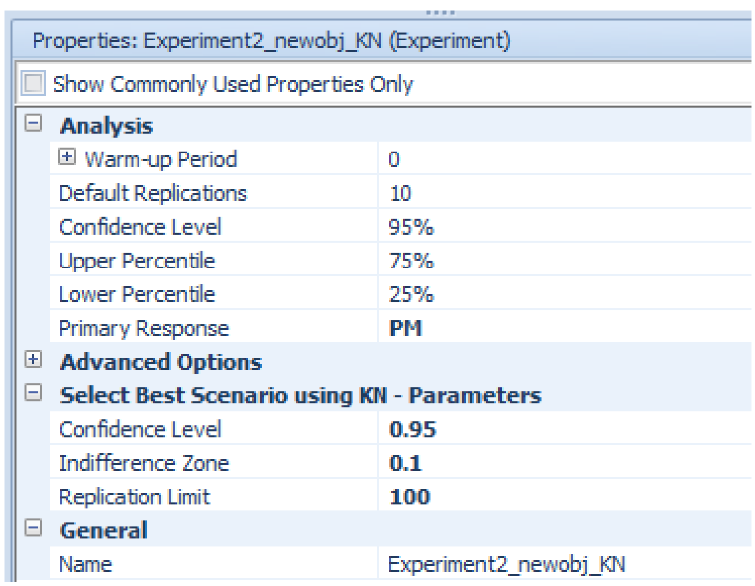

Figure 5.20: KN setting to select the best scenario set

The best solution set contains only one solution which we suggest it as the best solution

Table 5.8: The best solution

NoVisit	TimeOff(m)	TotalTimeOff%	AveTimeInSystem%	RunLength(h)	PM
3	9	22.41	29.88	69.84	4.27

for the problem. Hence, we suggest scheduling a time-off for each driver after three deliveries. Also, for a period of 69.84 (h) we suggest the time-off length equal to 9 (m) which is about 0.21% of total run length. In this situation, the total time-off percentage for each driver is about 22% and average time in system for each load is about 30% of total run length. Also, the value for proposed performance measure is about 4.27.

5.4 Conclusion

In this work, we simulate day-today operations of the considered truckload transportation network. We simulate two types of network which are point-to-point (PtP) and relay point network (RPN). Also, we investigate two types dispatching approach to assign the loads to the drivers. The method, namely, dispatcher-based dispatching approach (DbD) and a new method which we suggest in this study and call it collaborative dispatching approach (CD). In the latter case, the assignment decisions are made based on a collaboration between drivers and dispatcher in spit of DbD in which the dispatcher is the only decision maker.

We design a set of experiments to examine the performance of four different models achieved by two network types and two dispatching approaches, and suggest the best setting for the system through a simulation-based optimization. To do so, we define two input variables which are the number of load deliveries before scheduling a time-off for drivers and the length of time-off for each time-off. Also, we define two important measures or objectives which are the average total time-off for each driver and average time in system for each load. The first objective addresses the driver point of view such that the higher values for this measures increase the satisfactory level among the drivers (by increasing the total time-off and regularizing work schedule) and decrease the driver turnover rate. On the other hand, increasing the first objective meaninglessly can extend the total runtime and delivery times which is not appropriate from transportation companies perspective. Hence, we suggest the second objective to control this characteristic of the system. Regarding these two objectives we find a model which outperform the others. Afterward, to find the best values for our input values, controlling both objectives in an interesting level, we devise a simulation-optimization method based on multi-objective optimization approach. We also suggest a comprehensive performance measure which can be used as a single objective to optimize the design of the network.

In terms of computational results, we show that RP-network utilizing collaborative dispatching approach outperforms all the other models when it has an appropriate number of vehicles. Also, the best setting for this network is attained by scheduling a time-off for drivers after 3 deliveries whose length is about 0.21% of total run length of the model. We claim that using the suggested setting can decrease the driver turnover rate while make the truckload transportation companies satisfied.

Chapter 6

Conclusions and Future Research

This dissertation address the problem of high driver turnover rate in the TL transportation industry. We suggest utilizing the relay points through designing a well-structured relay network to alleviate this problem properly. RP-networks because of their structure can enhance the quality of life among the drivers. In this regard, we consider the design of a relay network having two special considerations, parallel shipment and multiple assignments generalizing the problem and decreasing the overall cost. Utilizing this new network setting, we can improve the satisfactory level for both drivers and transportation company, concurrently.

In the first work of this dissertation, we propose a mixed-integer programming formulation for the problem and propose an efficient solution approach based on Benders decomposition algorithm. Because of inefficiencies in the conventional Benders' decomposition to solve the proposed model, we employ several algorithmic enhancements including initial heuristics working based on a modified version of Dijkstra algorithm, a cut disaggregation scheme discussed, Bender's cut strengthening approach, surrogate constraints, etc. Our computational studies show that the performance of the proposed solution method is interesting for different classes of test instances such that on average its runtime is less than that of B&C, implemented by Cplex, by about 85% for those test instances which are solvable by both Cplex and proposed algorithm. More than 70% of large size instances are not solvable by Cplex. Also, the memory management of our algorithm is effective in such a way that it can manage the memory to solve the large size instances properly. We assess the problem and the algorithm in three different cases to show the performances of all parts of the solution method, separately. We also examine various problem and algorithm parameters on the algorithmic performance and solution characteristics for different cases. Furthermore, we present a comprehensive and in-detail report in terms of network characteristics and design information such as the required open RP locations, the direct shipment percentage among different cases, average number of visited RPs for each load, etc.

In the second work of this dissertation, we address the RP-network design problem under demand uncertainty. Uncertainties in the network's demands are inevitable in the transportation industry. We devise a framework to design an RP-network with uncertain or stochastic demands. Our problem assumptions and settings are set based on our work in the previous chapter of this dissertation. We suggest a two-stage stochastic program to model considered problem in a stochastic setting. To solve the problem, we propose a new version of the L-shaped method, namely, SC-based cut L-shaped method working based on a scenario categorization scheme introduced. We show that this version of L-shaped method can outperform two well-known versions of it, single-cut and multi-cut, to solve the proposed problem. Also, we suggest a number of enhancements for improving the performance of the proposed L-shaped method. The most important one is using a new version of the progressive hedging algorithm which we devise for SMIPs to generate a reliable initial solution for L-shaped. We show that this technique can improve the runtime, the number of iterations and the initial gap percentage of L-shaped. Other than this, we suggest including the mean value problem-based cut (MVP) to the L-shaped master problem. Also, we offer using the shortest path cuts to improve the initial lower bound of the algorithm and show that adding these cuts can improve the runtime of the algorithm. Adding disaggregated cuts and early termination the L-shaped master problem are two other enhancements we discussed to improve L-shaped performance. The computational results show the superiority of our solution method. Our overall algorithm, namely, PH-SLS, can improve the runtime and gap% of the single-cut L-shaped method by 40% and 37%, respectively.

In the third work of this dissertation, we simulate operations of the considered networks. We model two types of the network which are point-to-point (PtP) and relay point network (RPN). Also, we address two dispatching approach methods to assign loads to the drivers. The dispatcher-based dispatching approach (DbD) and a new method which we introduce in this study and call it collaborative dispatching approach (CD). In CD approach, the assignments are made based on a collaboration between drivers and dispatcher in spit of DbD approach in which the dispatcher is the only decision-maker. We design a set of experiments to examine the performance of four different models built by two network types and two dispatching approaches and suggest the best setting for the system through a simulation-based optimization. We devise a simulation-optimization method based on multi-objective optimization approach to optimize two different objective functions considering the drivers and the transportation company point of views, concurrently. We also suggest a comprehensive performance measure which can be used as a single objective to optimize the design of the network. In terms of computational results, we show that RP-network utilizing collaborative dispatching approach outperforms all the other models when it has an appropriate number of vehicles.

Considering capacitated relay point or arcs in the problem can be regarded as the immediate extensions of our study which can be addressed by incorporating required extensions and modifications to the proposed models. On the other hand, considering arc activation cost in the model is another direction of investigating the RR-network design problem which can be useful for other industries such as the telecommunication industry as well as the transportation industry. Also, in term of solution methodology, devising different approaches to solve the problem in the deterministic and stochastic environment can be considered as another opportunity to extend this work.

Bibliography

[1] ATA, "New report finds trucking industry revenues topped $700 billion," http://www.trucking.org/article/New-Report-Finds-Trucking-Industry-Revenues-Topped-$700-Billion, 2018. 1

[2] ATA-Business-Solutions, "Ata u.s. freight transportation forecast to 2028," https://www.atabusinesssolutions.com/ATA-Store/ProductDetails/productid/3614937, 2017. 1

[3] H. Min and T. Lambert, "Truck driver shortage revisited," *Transportation journal*, pp. 5–16, 2002. 1

[4] S. B. Keller, "Driver relationships with customers and driver turnover: key mediating variables affecting driver performance in the field," *Journal of Business Logistics*, vol. 23, no. 1, pp. 39–64, 2002. 1

[5] J. Mele, "Carriers cope with driver shortage," *Fleet Owner*, vol. 84, no. 1, pp. 104–111, 1989. 1

[6] ——, "Solving driver turnover," *Fleet Owner*, 1989. 1

[7] H. Richardson, "Can we afford the driver shortage?" *Transportation & Distribution*, 1994. 1

[8] ATA, "Turnover rate at large truckload carriers rises in first quarter," http://www.trucking.org/article/Turnover-Rate-at-Large-Truckload-Carriers-Rises-in-First-Quarter, 2018. 1

[9] P. Kewcharoenwong and H. Üster, "Relay network design with capacity and link-imbalance considerations: A lagrangean decomposition algorithm and analysis," *Transportation Science*, vol. 51, no. 4, pp. 1177–1195, 2017. 1, 3, 5, 9, 10

[10] J. Rodriguez, M. Kosir, B. Lantz, G. Griffin, and J. Glatt, "The costs of truckload driver turnover," *Fargo, Upper Great Plains Transportation Institute*, pp. 1–18, 2000. 1

[11] S. B. Keller and J. Ozment, "Managing driver retention: effects of the dispatcher," *Journal of Business Logistics*, vol. 20, no. 2, p. 97, 1999. 1

[12] G. D. Taylor, G. L. Whicker, and J. S. Usher, "Multi-zone dispatching in truckload trucking," *Transportation Research Part E: Logistics and Transportation Review*, vol. 37, no. 5, pp. 375–390, 2001. 2, 3, 8, 12

[13] ATA, "New report says national shortage of truck drivers to reach 50,000 this year," https://www.trucking.org/article/New%20Report%20Says-National-Shortage-of-Truck-Drivers-to-Reach-50,000-This-Year, 2017. 2

[14] H. Min and A. Emam, "Developing the profiles of truck drivers for their successful recruitment and retention: a data mining approach," *International Journal of Physical Distribution & Logistics Management*, vol. 33, no. 2, pp. 149–162, 2003. 2

[15] H. Üster and N. Maheshwari, "Strategic network design for multi-zone truckload shipments," *IIE transactions*, vol. 39, no. 2, pp. 177–189, 2007. 3, 5, 8, 10

[16] H. Üster and P. Kewcharoenwong, "Strategic design and analysis of a relay network in truckload transportation," *Transportation Science*, vol. 45, no. 4, pp. 505–523, 2011. 3, 5, 8, 10, 13, 25

[17] T. T. Taha and G. D. Taylor, "An integrated modeling framework for evaluating hub-and-spoke networks in truckload trucking," *Logistics and Transportation Review*, vol. 30, no. 2, p. 141, 1994. 8, 12

[18] G. D. Taylor, S. Harit, J. R. English, and G. Whicker, "Hub and spoke networks in truckload trucking: Configuration, testing and operational concerns," *Logistics and Transportation Review*, vol. 31, no. 3, p. 209, 1995. 8

[19] G. D. Taylor, T. Meinert, R. Killian, and G. Whicker, "Development and analysis of alternative dispatching methods in truckload trucking," *Transportation Research Part E: Logistics and Transportation Review*, vol. 35, no. 3, pp. 191–205, 1999. 8

[20] G. D. Taylor and T. S. Meinert, "Improving the quality of operations in truckload trucking," *Iie Transactions*, vol. 32, no. 6, pp. 551–562, 2000. 8

[21] G. Hunt, "Transportation relay network design. atlanta, ga," Ph.D. dissertation, Ph. D. Thesis, Georgia Institute of Technology, 1998. 8

[22] T. H. Ali, S. Radhakrishnan, S. Pulat, and N. C. Gaddipati, "Relay network design in freight transportation systems," *Transportation Research Part E: Logistics and Transportation Review*, vol. 38, no. 6, pp. 405–422, 2002. 8

[23] K. D. Melton and R. Ingalls, "Utilizing relay points to improve the truckload driving job," *International Journal of Supply Chain Management*, vol. 1, no. 3, 2013. 8

[24] H. A. Vergara and S. Root, "Mixed fleet dispatching in truckload relay network design optimization," *Transportation Research Part E: Logistics and Transportation Review*, vol. 54, pp. 32–49, 2013. 9

[25] E. A. Cabral, E. Erkut, G. Laporte, and R. A. Patterson, "The network design problem with relays," *European Journal of Operational Research*, vol. 180, no. 2, pp. 834–844, 2007. 9

[26] M. Leitner, I. Ljubić, M. Riedler, and M. Ruthmair, "Exact approaches for network design problems with relays," Technical Report AC-TR-15-003, Algorithms and Complexity Group, TU Wien, Tech. Rep., 2015. 9

[27] B. Yıldız, O. E. Karaşan, and H. Yaman, "Branch-and-price approaches for the network design problem with relays," *Computers & Operations Research*, vol. 92, pp. 155–169, 2018. 9

[28] S. Alumur and B. Y. Kara, "Network hub location problems: The state of the art," *European journal of operational research*, vol. 190, no. 1, pp. 1–21, 2008. 9

[29] 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*, vol. 64, no. 4, pp. 1096–1109, 2013. 9

[30] J. F. Campbell and M. E. O'Kelly, "Twenty-five years of hub location research," *Transportation Science*, vol. 46, no. 2, pp. 153–169, 2012. 9, 10

[31] J. Campbell, A. Ernst, and M. Krishnamoorthy, "Hub location problems." *Drezner, Z; Hamacher, H editors, editor/s. Facility location : application and theory*, pp. 373–407, 2002. 9

[32] S. A. Alumur, B. Y. Kara, and O. E. Karasan, "The design of single allocation incomplete hub networks," *Transportation Research Part B: Methodological*, vol. 43, no. 10, pp. 936–951, 2009. 10

[33] H. Calık, S. A. Alumur, B. Y. Kara, and O. E. Karasan, "A tabu-search based heuristic for the hub covering problem over incomplete hub networks," *Computers & Operations Research*, vol. 36, no. 12, pp. 3088–3096, 2009. 10

[34] R. S. de Camargo, G. de Miranda J., M. E. O'Kelly, and J. F. Campbell, "Formulations and decomposition methods for the incomplete hub location network design problem with and without hop-constraints," *Applied Mathematical Modelling*, vol. 51, pp. 274–301, 2017. 10

[35] M. Riis and K. A. Andersen, "Capacitated network design with uncertain demand," *INFORMS Journal on Computing*, vol. 14, no. 3, pp. 247–260, 2002. 10

[36] A. Atamtürk and M. Zhang, "Two-stage robust network flow and design under demand uncertainty," *Operations Research*, vol. 55, no. 4, pp. 662–673, 2007. 10

[37] R. Rahmaniani, T. G. Crainic, M. Gendreau, and W. Rei, "Accelerating the benders decomposition method: Application to stochastic network design problems," *SIAM Journal on Optimization*, vol. 28, no. 1, pp. 875–903, 2018. 11

[38] R. T. Rockafellar and R. J.-B. Wets, "Scenarios and policy aggregation in optimization under uncertainty," *Mathematics of operations research*, vol. 16, no. 1, pp. 119–147, 1991. 11, 63

[39] A. Løkketangen and D. L. Woodruff, "Progressive hedging and tabu search applied to mixed integer (0, 1) multistage stochastic programming," *Journal of Heuristics*, vol. 2, no. 2, pp. 111–128, 1996. 11, 63

[40] O. Listes and R. Dekker, "A scenario aggregation–based approach for determining a robust airline fleet composition for dynamic capacity allocation," *Transportation Science*, vol. 39, no. 3, pp. 367–382, 2005. 11, 63

[41] J. P. Watson and D. L. Woodruff, "Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems," *Computational Management Science*, vol. 8, no. 4, pp. 355–370, 2011. 11, 63

[42] G. Guo, G. Hackebeil, S. M. Ryan, J. P. Watson, and D. L. Woodruff, "Integration of progressive hedging and dual decomposition in stochastic integer programs," *Operations Research Letters*, vol. 43, no. 3, pp. 311–316, 2015. 11

[43] D. Gade, G. Hackebeil, S. M. Ryan, J. P. Watson, R. J.-B. Wets, and D. L. Woodruff, "Obtaining lower bounds from the progressive hedging algorithm for stochastic mixed-integer programs," *Mathematical Programming*, vol. 157, no. 1, pp. 47–67, 2016. 11, 67

[44] Y. Fan and C. Liu, "Solving stochastic transportation network protection problems using the progressive hedging-based method," *Networks and Spatial Economics*, vol. 10, no. 2, pp. 193–208, 2010. 11, 63

[45] T. G. Crainic, X. Fu, M. Gendreau, W. Rei, and S. W. Wallace, "Progressive hedging-based metaheuristics for stochastic network design," *Networks*, vol. 58, no. 2, pp. 114–124, 2011. 11

[46] T. G. Crainic, M. Hewitt, and W. Rei, "Scenario grouping in a progressive hedging-based meta-heuristic for stochastic network design," *Computers & Operations Research*, vol. 43, pp. 90–99, 2014. 11

[47] T. T. Taha, G. D. Taylor, and H. A. Taha, "A simulation-based software system for evaluating hub-and-spoke transportation networks," *Simulation practice and theory*, vol. 3, no. 6, pp. 327–346, 1996. 12

[48] T. L. Magnanti and R. T. Wong, "Accelerating benders decomposition: Algorithmic enhancement and model selection criteria," *Operations research*, vol. 29, no. 3, pp. 464–484, 1981. 21, 56

[49] T. J. Van Roy, "A cross decomposition algorithm for capacitated facility location," *Operations Research*, vol. 34, no. 1, pp. 145–163, 1986. 21, 56

[50] H. Üster and H. Agrahari, "A benders decomposition approach for a distribution network design problem with consolidation and capacity considerations," *Operations Research Letters*, vol. 39, no. 2, pp. 138–143, 2011. 21, 56

[51] R. M. Van Slyke and R. Wets, "L-shaped linear programs with applications to optimal control and stochastic programming," *SIAM Journal on Applied Mathematics*, vol. 17, no. 4, pp. 638–663, 1969. 53

[52] H. Üster and S. O. Hwang, "Closed-loop supply chain network design under demand and return uncertainty," *Transportation Science*, vol. 51, no. 4, pp. 1063–1085, 2016. 60

[53] S. Batun, B. T. Denton, T. R. Huschka, and A. J. Schaefer, "Operating room pooling and parallel surgery processing under uncertainty," *INFORMS journal on Computing*, vol. 23, no. 2, pp. 220–237, 2011. 60

[54] J. L. W. V. Jensen *et al.*, "Sur les fonctions convexes et les inégalités entre les valeurs moyennes," *Acta mathematica*, vol. 30, pp. 175–193, 1906. 60