Analysis and Optimization of Chassis Movements in Transportation Networks with Centralized Chassis Processing Facilities

Timothy Martin VanderBeek
Claremont Graduate University

Follow this and additional works at: https://scholarship.claremont.edu/cgu_etd

Part of the Applied Mathematics Commons, and the Transportation Engineering Commons

Recommended Citation
https://scholarship.claremont.edu/cgu_etd/505.

This Open Access Dissertation is brought to you for free and open access by the CGU Student Scholarship at Scholarship @ Claremont. It has been accepted for inclusion in CGU Theses & Dissertations by an authorized administrator of Scholarship @ Claremont. For more information, please contact scholarship@cuc.claremont.edu.
Analysis and Optimization of Chassis Movements in Transportation Networks

with Centralized Chassis Processing Facilities

by

Tim VanderBeek

Claremont Graduate University and California State University Long Beach

2019

© Copyright Tim VanderBeek, 2019 All rights reserved
APPROVAL OF THE REVIEW COMMITTEE

The dissertation has been duly read, reviewed, and critiqued by the Committee listed below, which hereby approves the manuscript of Tim VanderBeek as fulfilling the scope and quality requirements for meriting the degree of Engineering and Industrial Applied Mathematics.

Dr. Anastasios Chassiakos, Chair
California State University Long Beach
Professor

Dr. Hossein Jula
California State University Long Beach
Professor

Dr. Allon Percus
Claremont Graduate University
Professor

Dr. Henry Schellhorn
Claremont Graduate University
Professor
Abstract

Analysis and Optimization of Chassis Movements in Transportation Networks with Centralized Chassis Processing Facilities

by

Tim VanderBeek

Claremont Graduate University and California State University Long Beach: 2019

This work studies the concept of “Centralized Processing of Chassis,” and its potential impact on port drayage efficiency. The concept revolves around an off-dock terminal (or several off-dock terminals), referred to as Chassis Processing Facilities (CPF). A CPF is located close to the port, where trucks will go to exchange chassis, thereby reducing traffic at the marine terminals and resulting in reduced travel times and reduced congestion. This work is divided into two major studies: one at the strategic planning level, and one at the operational level for individual trucking companies.

In the first study, an analytical framework for modeling and optimization of chassis movements in transportation networks with CPFs is developed, and a case study in the Long Beach/Los Angeles (LB/LA) port area is performed. Comparisons between current practices at ports, in which chassis exchanges occur at marine terminals, and proposed practices, in which the exchanges happen at CPFs, are performed. The results of this study indicate that a reduction of total travel time by up to 20% can be achieved when using the CPFs. The study also shows that, in the LB/LA port area, the return on investment for establishing additional CPF locations decreases sharply for any more than three CPFs. Overall, the findings indicate that travel time can be significantly reduced through implementation of CPFs which has important implications
in reducing negative environmental impacts of the port as well as operational costs for trucking companies.

In the second study, scheduling of chassis and container movements is optimized at the operational level for individual trucking companies, when CPFs are available for use within a major metropolitan area. A multi-objective optimization problem is formulated in which the weighted combination of the total travel time for the schedules of all vehicles in the company fleet and the maximum work span across all vehicle drivers during the day is minimized. Time-varying dynamic models for the movements of chassis and containers are developed and used in the optimization process. The optimal solution is obtained through a genetic algorithm, and the effectiveness of the developed methodology is evaluated through a case study which once again focuses on the LB/LA port area. The case study uses a trucking company located in the Los Angeles region, which can utilize three candidate CPFs for exchange of chassis. The company assigns container movement tasks to its fleet of trucks, with warehouse locations spread across the region. In the simulation scenarios developed for the case study, the use of CPFs at the trucking company level, can provide improvements up to 30% (depending upon the specific scenario) over the cases not using any CPFs. It is found in this work that for typical cases where the number of jobs is much larger than the number of vehicles in the company fleet, the greatest benefit from CPF use would be in the cases where there are some significant job-to-job differences with respect to chassis usage and type.

Lastly, in addition to the formulation and optimization for initially planning daily activities, the study further models the problem in a dynamic environment, in which traffic network parameters can change drastically from initial daily predictions. In order to perform the optimization in a dynamic formulation with varying noise levels, a method by which noise could be injected into
the initial daily predictions is developed to support the model inputs for the case study and an incremental optimization approach is implemented. Results indicate that a modest potential benefit of approximately 2% may be expected if dynamic re-routing is performed. However, in practice it will be important to weigh the cost of the additional real-time queries required to enable the dynamic re-routing against the potential benefits for the specific company and job set in question prior to implementation.
Dedication

To

Jolene, Lottie and Lawson,

I love you to the sun and back
Acknowledgements

Throughout my academic career, I have been fortunate to have a community of family, friends, scholars, and colleagues who have supported me. Without them I would never have made it this far.

I am extremely appreciative of my dissertation committee: Dr. Anastasios Chassiakos, Dr. Hossein Jula, Dr. Allon Percus, and Dr. Henry Schellhorn. The advice, feedback and mentorship I received during my time as a PhD student and, in particular, during the process of completing my dissertation was invaluable. I am grateful to Anastasios and Hossein for providing critical feedback and pushing me to reach further, and I truly appreciate Allon and Henry’s encouragement and mentorship which helped me to grow my research skills and confidence. In addition, I would like to thank the National Center for Metropolitan Transportation Research (METRANS) located at the University of Southern California (USC) and the California State University Long Beach (CSULB) as well as the U.S. Department of Transportation who supported this work.

I would also like to thank my colleagues and mentors with whom I have been lucky to work throughout my years at Boeing, especially, Wayne Martin, Bruce Pope, Greg Lepore, Mike Albertson, Harry Henshaw, Jeff Strunk, Ray Schmidt, Dr. Dwayne Heckman, Dr. Chan Park, Marlene Ventresco, and Dennis Matsumoto. Since the outset of my career, I was surrounded by these exceptional engineers and mathematicians at the top of their field. Learning from them and seeing the amazing things they could accomplish helped foster in me a desire to continue to further my education. I would also like to thank my colleagues at CSULB, especially, Justin Nielsen, Casey Blonski, and Greg Capps. Early in my collegiate career, I was very fortunate to meet and work with such bright, hard-working, and articulate engineers who were a joy to solve problems with.

As for my family, I first need to thank my parents, Jerry and Pat VanderBeek, for their love of mathematics which was a constant presence growing up. Even before I was born, my life was destined to be filled with math, as they met in a math class at UC Davis. Since then they supported me at home,
taught me at school, and helped point me in the direction of interesting career paths that would allow me to do math for a living. They helped guide my academic career from the outset to lead me to where I am today. I also would like to thank my brothers Nathan and Ben for being academic role models and providing me with support throughout my life. In particular, I remember Jolene and I, just having returned from the Peace Corps, were living at Nathan’s house when I first decided to apply to PhD programs.

I am so very blessed by and thankful to my children Lottie and Lawson. Thank you for understanding when I had to be away, and motivating me to get things done so I could come back to you. And most of all thank you to Dr. Jolene McCall. You never cease to amaze me with all that you can do and all that you can achieve. You hold our family together, keep us moving and make me look forward to new challenges and adventures. You have always supported me, and believed in me and my capabilities. You bring clarity when I need it most. If I could, I’d spend every moment with you until you get tired of seeing my face. There is no greater joy in my life than spending time with you and our family, and with this milestone behind me, I look forward to a lot more of it in the future.
# TABLE OF CONTENTS

1 INTRODUCTION .......................................................................................................................... 15
1.1 Drayage and Container Handling Operational Efficiency .................................................. 18
1.2 Typical Transaction Types for Container Transport ......................................................... 21
1.3 Centralized Processing of Chassis ...................................................................................... 24

2 CHASSIS PROCESSING LOCATION OPTIMIZATION .............................................................. 26
2.1 Analytical Models and Optimization .................................................................................. 27
2.1.1 Import Only with Unlimited Chassis Processing Facility Usage ............................. 28
2.1.2 Import Only with Limited Chassis Processing Facility Usage ............................... 31
2.1.3 Export and Import Formulation ................................................................................. 33
2.2 Case Study Model Implementation ................................................................................. 37
2.2.1 Marine Terminals ......................................................................................................... 39
2.2.2 Trucking Companies ..................................................................................................... 42
2.2.3 Central Processing Facilities ...................................................................................... 42
2.2.4 Travel Time Between Locations .................................................................................. 44
2.2.5 Improved Chassis Processing Times at CPFs ............................................................ 45
2.3 Case Study Simulation Results ......................................................................................... 46
2.3.1 Optimization Software ................................................................................................. 46
2.3.2 Import Only with Unlimited CPF Usage ................................................................. 48
2.3.2.1 Reduced-Node Model Evaluation: Six Nodes .................................................. 48
2.3.2.2 Reduced-Node Model Evaluation: Ten Nodes ............................................... 52
2.3.2.3 Full Model (101-node model) ................................................................................ 58
2.3.3 Import Only Transactions with Restricted CPF Availability ..................................... 61
2.3.4 Export/Import Integer Linear Formulation with the Full Model ............................. 64
2.3.5 Linear Program Sensitivity Analysis ......................................................................... 73
2.4 Summary ............................................................................................................................... 79

3 DYNAMIC DRAYAGE TRUCK SCHEDULING USING CPFS ..................................................... 81
3.1 Analytical Models and Optimization .................................................................................. 85
3.1.1 Problem Formulation ...................................................................................................... 86
3.1.2 Optimization Methodology .......................................................................................... 92
3.1.3 Genetic Algorithm Overview ...................................................................................... 93
3.1.4 Initial Population ......................................................................................................... 95
3.1.5 Fitness Function .......................................................................................................... 100
3.1.6 Crossover Function .................................................................................................... 100
3.1.7 Mutation Function .................................................................................................... 101
3.1.8 Termination Function ................................................................................................ 104
3.2 Case Study Model Implementation ............................................................................... 104
3.2.1 Jobs ............................................................................................................................. 105
3.2.2 Travel Time Between Locations ............................................................................... 106
3.2.2.1 Single Sample per Route Least Squares Model .............................................. 107
3.2.2.2 Multiple Sample per Route Least Squares Model .......................................... 125
3.2.2.3 Adding Noise to the Model ................................................................................. 145
3.2.2.4 Triangle Fixing ...................................................................................................... 150
3.2.2.5 Adding Accidents ................................................................................................. 153
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.2.6</td>
<td>Modeling Change in Prediction Accuracies</td>
<td>161</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Additional Time Settings</td>
<td>162</td>
</tr>
<tr>
<td>3.2.4</td>
<td>Optimization and Genetic Algorithm Specific Settings</td>
<td>163</td>
</tr>
<tr>
<td>3.3</td>
<td>Case Study Simulation Results</td>
<td>168</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Genetic Algorithm Evaluation</td>
<td>168</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Impact of CPFs</td>
<td>172</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Sensitivity Analysis for $\mu$</td>
<td>182</td>
</tr>
<tr>
<td>3.3.4</td>
<td>Impact of time-varying model route accuracy</td>
<td>184</td>
</tr>
<tr>
<td>3.4</td>
<td>Summary</td>
<td>190</td>
</tr>
<tr>
<td>4</td>
<td>Conclusions</td>
<td>192</td>
</tr>
<tr>
<td>5</td>
<td>References</td>
<td>194</td>
</tr>
</tbody>
</table>
FIGURES

Figure 1. World’s container port traffic in TEU (2000-2017) .......................................................... 15
Figure 2. Container port traffic in TEU by country (2017). ............................................................ 16
Figure 3. Annual TEU throughput at POLB & POLA (1997-2017). ............................................. 17
Figure 4. Annual change in TEU throughput for the combined ports (2012-2017). ..................... 17
Figure 5. Description of container transaction types at MTs .......................................................... 23
Figure 6. Schematic of Centralized Processing of Chassis concept .............................................. 27
Figure 7. Node locations for the full model used in the simulation .............................................. 39
Figure 8. POLB & POLA terminal locations ............................................................................. 41
Figure 9. Example of chassis stacking layout within a CPF .......................................................... 43
Figure 10. Chassis pick-up process ............................................................................................... 45
Figure 11. Centralized Processing of Chassis example ............................................................... 50
Figure 12. Optimal transaction routing vs. parameter P (six-node case) ..................................... 52
Figure 13. Optimal transaction routing vs. parameter P (ten-node case) ..................................... 57
Figure 14. Number of transactions routed through CPFs at optimality, as a function of P ....... 59
Figure 15. Optimal transaction routing vs. parameter P: one CPF available ......................... 62
Figure 16. Optimal value of the objective function vs. parameter P: one CPF available .......... 63
Figure 17. Number of CPFs used vs. total travel time ................................................................. 65
Figure 18. Map of the optimal CPF locations (P = 1,200 seconds) ......................................... 67
Figure 19. Map of the optimal CPF locations (P = 1,200 seconds) ......................................... 67
Figure 20. Ranking of CPF candidate locations (P = 600 seconds) ....................................... 72
Figure 21. Percent change in total travel time vs. ratio of (export/import) transactions .......... 79
Figure 22. Example schematic of vehicle routing problem with CPFs ....................................... 83
Figure 23. General process flow for vehicle schedule ................................................................. 85
Figure 24. Example of chromosomes and genes used in the genetic algorithm ...................... 93
Figure 25. Genetic algorithm overview ....................................................................................... 94
Figure 26. Nearest Neighbor Algorithm 1 example output ...................................................... 96
Figure 27. Nearest Neighbor Algorithm 2 example output ...................................................... 97
Figure 28. Random Permutation Algorithm 1 example output ............................................... 98
Figure 29. Random Permutation Algorithm 2 example output ............................................... 99
Figure 30. Crossover example .................................................................................................. 101
Figure 31. Mutation Function 1 example .................................................................................... 102
Figure 32. Mutation Function 2 example .................................................................................... 102
Figure 33. Mutation Function 3 example .................................................................................... 103
Figure 34. Map of jobs used in case study .................................................................................. 106
Figure 35. Map of jobs used in daily traffic variation model .................................................... 109
Figure 36. Peak predicted travel time (recommended route is highlighted in blue) ............... 111
Figure 37. Daily travel variation optimistic / pessimistic estimates (Trip 4) .............................. 112
Figure 38. Optimistic travel time estimates ................................................................................. 113
Figure 39. Pessimistic travel time estimates ............................................................................... 113
Figure 40. Average travel time estimates ..................................................................................... 114
Figure 41. Daily travel variation vs. least squares fit (1-2) ......................................................... 117
Figure 42. Daily travel variation vs. least squares fit (3-4) ......................................................... 118
Figure 43. Daily travel variation vs. least squares fit (5-6) ......................................................... 119
Figure 44. Daily travel variation vs. least squares fit (7-8) ......................................................... 120
Figure 45. Daily travel variation vs. least squares fit (9-10) ...................................................... 121
Figure 92. Scenario 2: Solution (a) w/o CPFs and (b) w/ CPFs, N = 15, P = 1,200 seconds .... 180
Figure 93. Scenario 2: Solution (a) w/o CPFs and (b) w/ CPFs, N = 60, P = 1,200 seconds .... 181
Figure 94. Total travel time vs. $\mu$. ........................................................................... 183
Figure 95. Work span vs. $\mu$............................................................................... 183
Figure 96. Percent degradation in solution due to errors in predicted travel durations. .......... 185
Figure 97. Route duration error magnitudes vs. time. .................................................. 186
Figure 98. Route duration error magnitudes vs. time with accidents. .............................. 187
Figure 99. Average route duration error magnitudes..................................................... 188
Figure 100. Percent improvement in solution with real-time dynamic re-routing. .......... 190

TABLES

Table 1. Locations of POLB and POLA terminals used in the case study. ......................... 40
Table 2. POLB and POLA export and import statistics for 2017.................................... 41
Table 3. Potential CPF locations and capacities for chassis storage............................... 44
Table 4. Travel times (in seconds) from TCs to CPFs (six-node case). ......................... 49
Table 5. Travel times (in seconds) from CPFs to TCs (six-node case). ............................. 49
Table 6. Travel Times (in seconds) from TCs to MTs (six-node case). ............................. 49
Table 7. Optimal transaction routing (six-node case). .................................................. 51
Table 8. Travel times (in seconds) from TCs to CPFs (ten-node case). .............................. 54
Table 9. Travel times (in seconds) from CPFs to TCs (ten-node case). .............................. 54
Table 10. Travel times (in seconds) from TCs to MTs (ten-node case). ............................. 55
Table 11. Optimal transaction routing (ten-node case). ................................................ 56
Table 12. CPF capacity utilization (P=1,200 seconds). ................................................. 60
Table 13. Ranking of potential CPF locations if only CFP is available. .......................... 64
Table 14. Top CPF ranking vs. number of CPFs used: $P = 1,200$ seconds ..................... 68
Table 15. Top CPF ranking vs. number of CPFs used: $P = 600$ seconds ....................... 70
Table 16. Top CPF ranking export/import.......................................................................... 73
Table 17. CPF rankings vs. CPF capacity: import transactions only .............................. 75
Table 18. CPF rankings vs. CPF capacity: export and import transactions ..................... 77
Table 19. CPF rankings vs. total number of transactions: export and import transactions. .. 78
Table 20. Schedule example for vehicle $V_m$................................................................ 84
Table 21. Origin destination pairs for daily variation estimates........................................ 108
Table 22. Optimal sample times vs. sample quantity. .................................................... 130
1 INTRODUCTION

The volume of intermodal containerized freight transportation continues to increase, as world ports handled more than 750 billion twenty-foot equivalent units (TEUs) of containers in 2017 (Figure 1 and Figure 2). This is more than a 200% increase since the year 2000, and this number is only expected to grow higher in the future (The World Bank, 2018).

Figure 1. World’s container port traffic in TEU (2000-2017).
Figure 2. Container port traffic in TEU by country (2017).

In particular, the twin ports of Long Beach (POLB) and Los Angeles (POLA), consisting of fourteen individually gated terminals, combine to create the largest container port complex in the US, having handled 16.9 million TEUs in 2017 (Port of Long Beach Statistics, 2018), (Port of Los Angeles Statistics, 2018), (The Tioga Group, 2002). Since most of the containers in use are forty-foot units, the figure of 16.9 million TEUs corresponds to approximately 9.1 million individual container units (the conversion factor most widely used in the industry is: One Individual Container = 1.85 TEU, (The Tioga Group, 2002)). Figure 3 shows the annual TEU throughput at the POLB and POLA for the period 1997-2017 (Port of Long Beach Statistics, 2018), (Port of Los Angeles Statistics, 2018). Although the explosive growth of the first ten years exhibited a slowdown after the recession of 2008, it has recovered in the last few years, reaching its pre-recession levels. The numbers in Figure 3 include both loaded and empty units, destined for export or import. Figure 4 shows the change in total annual TEU throughput for the combined ports. The yearly change over the last six years is positive. The total container throughput
(export and import) through the POLB and POLA is expected to grow in the future, correlated with population increase, domestic demand for inexpensive manufactured goods, as well as global demand for US agricultural products, and improving competitiveness of US industry. Handling a large number of the necessary container transactions requires intensive management of operations, changes in transportation policy and modernized equipment.

Figure 3. Annual TEU throughput at POLB & POLA (1997-2017).

Figure 4. Annual change in TEU throughput for the combined ports (2012-2017).
Within the containerized intermodal transportation chain, the drayage operation, in which trucks transport containers between marine terminals (MTs) and customers, accounts for more than a quarter of the origin to destination expenses (Macharis & Bontekoning, 2004). In addition, the large volume of drayage operations impacts the areas around and within the ports through traffic congestion, noise pollution, and greenhouse gas emissions (Konstantzos, Saharidis, & Loizidou, 2017). Therefore, any method by which the drayage operations’ efficiency can be improved will have a positive impact on both the economy and the local environment.

One such concept which could improve the efficiency of drayage operations is the Centralized Processing of Chassis, in which off-dock terminals are implemented in the exchange of the chassis used to haul the containerized freight. Whereas the majority of the international community uses container chassis which are supplied by customers, truckers, or off-terminal pools, international chassis within the US are typically located at the MT, where limited space, increased queue times, and logistical complexities associated with chassis ownership can result in inefficiencies (The Tioga Group, Research, & Engineering, 2011), (Izmirli, 2015), (Bailey, Mollie; Hewitt, Sheila, 2014). Therefore, drayage operations’ efficiency may be improved if the chassis exchanges could instead occur away from the MT at off-dock locations, referred to herein as Chassis Processing Facilities (CPF), which do not share the same chassis exchange inefficiencies that currently exist at the MTs.

1.1 Drayage and Container Handling Operational Efficiency

Research problems in the areas of improved container handling and operational efficiency have received much attention from the academic community in the last several decades, as terminal operators, shipping lines and port authorities seek to make the most of their limited resources when implementing changes in the areas of new technologies and alternative approaches for managing containerized goods (Gharehgozli, Roy, & De Koster, 2016). Specific to drayage operations’ efficiency, one area which has gotten a lot of focus is the planning of gate operations and truck appointment systems, where various case studies have
shown that appointment systems have the potential to reduce congestion within the terminal (Shiri & Huynh, 2016). In particular, (Namboothiri & Erera, 2008) looked at minimizing transportation costs with a fleet of trucks based upon an integer programming heuristic, examining vehicle productivity as a function of available appointment slot capacity. (Chen & Yang, 2010) proposed partitioning truck arrivals on a ship specific basis with a two-step approach: using integer programming to optimize time windows, followed by implementing a genetic algorithm heuristic to generate a partitioning solution. Later, (Chen, Govindan, & Golias, 2013) also used a genetic algorithm when proposing a method to optimize truck arrival patterns and reduce emissions from idling truck engines. (Zehendner & Feillet, 2014) looked at leveraging the truck appointment system to not only increase the service quality of trucks, but also of trains, barges and vessels; and recently (Torkjazi, Huynh, & Shiri, 2018) generated a nonlinear program to model the potential benefits of optimizing truck arrival uniformity throughout the day, which resulted in additional drayage operational cost reduction as compared to an appointment system which only minimizes gate queuing times. However, little work has been done in the area of modeling the chassis related operations, and, in general, terminal appointments systems have had mixed performance; where (Giuliano & O’Brien, 2007), in particular, reported unsuccessful application at the POLB and POLA (Gharehgozli, Roy, & De Koster, 2016).

Additional optimization approaches for off-dock truck operations were examined by (Meisel & Kopfer, 2014), which dealt with the synchronization of resources (e.g. vehicles / people) used for passive items in the supply chain (e.g. support equipment for loading and unloading containers) with the active transportation elements (e.g. truck bobtails moving cargo from place to place). Similarly, (Zhaojie, Canrong, Wei-Hua, Lixin, & Peng, 2014) and (Zhaojie, Wei-Hua, Lixin, & Canrong, 2015) focused on the need to consider the separation of the tractor and trailer and optimization of truck scheduling with multiple vehicles in service and varying wait / processing times while allowing for different trucks to perform various aspects of the interface with containers. (Yujian, Jiantong, Zhe, & Chunming, 2017) also looked at aspects of the truck container operations, but this time modeled container maintenance, while
(Christian, Nicola, Michael, Stefan, & Frank, 2018) further analyzed the optimization of active and passive elements in the supply chain. However, while these various vehicle routing problems have examined elements of chassis and container usage, none of them have considered the potential benefits associated with centralization of the chassis operations.

Another area of research for improving drayage efficiency includes potential changes to the way MT container operations are performed. A large set of studies have been performed on specific details such as the modification of container stacking approaches. Papers have studied the effect of various layout variables and approaches on the performance of the terminal (Kim, Park, & Jin, 2008) (Gue, Ivanović, & Meller, 2012) (Wiese, Suhl, & Kliewer, 2013) (Lee & Kim, 2013). Some alternative container stacking layouts with smaller stacks and different sizes have achieved a reduction in vehicle travel time of up to 20% (Öztürkoğlu, Gue, & Meller, 2012). While this illustrates some of the improvements which can be accomplished within the terminal, container terminals can also transform their supply chains to increase their competitiveness and robustness by collaborating with hinterland terminals to balance flows and workload more efficiently over time (Heinrich, 2003) (Vervest & Zheng, 2009). As a result, some of the container operations can be relegated away from the container terminals, such as container inspection and container delivery to end customers. (Dekker, van der Heide, van Asperen, & Ypsilantis, 2013) built upon this concept in Europe by introducing the concept of the Chassis Exchange Terminal (CET) as a method for improving travel times associated with container retrieval. In the CET concept, the centralized processing of chassis occurred at an off-dock terminal (or a number of off-dock terminals) located close to the port, where trucks would go to retrieve imports or drop off exports instead of unloading and loading containers at the MT. However, the CET study was specific to the potential benefits due to operations at off-peak hours, for example at night time, and looked only at wheeled transactions in which containers are already loaded on a chassis and can simply be picked up or dropped off by the bobtail.

Although significant research up to this point has delved into potential methods by which the drayage operations’ efficiency could be improved, there is potential for additional gains through chassis utilization
and optimization using the CPF concept while taking into account the current operational scenarios and typical transaction types within the US.

1.2 Typical Transaction Types for Container Transport

In order to complete the export/import operations moving containers to/from MTs the transporting trucks will perform a series of steps including: dropping off export containers; dropping off empty chassis used for exports; picking up chassis for imports; picking up import containers; and traveling between any locations necessary to complete these tasks (O'Brien, Reeb, & Kunitsa, 2014), (Le-Griffin, Mai, & Griffin, 2011). The most common transaction types for trucking companies at MTs are listed below.

Type 1: Single transaction export
Type 2: Single transaction import of grounded container
(i.e. container not loaded on a chassis)
Type 3: Single transaction import of wheeled container
(i.e. container already loaded on chassis)
Type 4: Dual transaction export/import of grounded import
Type 5: Dual transaction export/import of wheeled import

Figure 5 shows the flow of bobtails, chassis and containers for transaction Types 1-5 described above.

The flows presented in Figure 5 depict the operations taking place between the in-gate and out-gate of the MT. The truck’s point of origin or its final destination, e.g. a warehouse or a parking space at the trucking company, are not depicted in the figure. The following list provides a detailed explanation of the operations taking place for each type of the five transactions.
• Type 1: Single transaction export. The bobtail leaves the trucking company (or its point of origin) with a chassis on which an export container is loaded. It arrives at the in-gate; enters the terminal; drops off the export container and the chassis in the MT; passes through the out-gate and arrives at its final destination as a bobtail.

• Type 2: Single transaction import of grounded container. The bobtail arrives at the in-gate; picks up a chassis at the MT; picks up an import container; passes through the out-gate and arrives at its final destination as a bobtail with a chassis and a container.

• Type 3: Single transaction import of wheeled container. The bobtail arrives at the in-gate; picks up a chassis which has already been loaded with an import container; passes through the out-gate and arrives at its final destination as a bobtail with a chassis and a container.

• Type 4: Dual transaction export/import of grounded import. The bobtail arrives at the in-gate with a chassis on which an export container is loaded; enters the terminal; drops off the export container; loads an import container to the chassis; passes through the out-gate and arrives at its final destination as a bobtail with a chassis and a container.

• Type 5: Dual transaction export/import of wheeled import. The bobtail arrives at the in-gate with a chassis on which an export container is loaded; enters the terminal; drops off the export container; drops off the chassis; picks up a chassis which has already been loaded with an import container; passes through the out-gate and arrives at its final destination as a bobtail with a chassis and a container.
Figure 5. Description of container transaction types at MTs.

Among the types of transactions described above (Figure 5) the Type 1 and Type 2 transactions are the only types which would be anticipated to contribute to a noticeable reduction in total transaction time if a CPF was used. In the case of Type 1 transactions, the export container can be dropped off at the desired MT, and then the chassis can be returned to the CPF for storage and later retrieval. In the case of a Type 2 transaction, the chassis for import can be picked up at the CPF before entering the MT to load the import container. In both cases if the chassis exchange transaction performed outside of the MT can be done more efficiently (where it is anticipated that there will be less congestion), this could offer improvements in total time for the transaction. In Type 3 and 4 transactions one can see that no chassis exchange activities are necessary. In a Type 3 transaction the wheeled import includes a container already loaded on a chassis and can simply be picked up by the bobtail. In a Type 4 transaction the chassis used for the export container is the same one onto which the import container can be loaded afterwards. Finally, Type 5 transactions, although they include a chassis exchange, would not be anticipated to have any reduced
transaction times using an external CPF. This is due to the fact that after dropping off an export, the bobtail must drop off the chassis used with the export prior to picking up a wheeled import at the same terminal, making it inefficient to travel to an external CPF to drop off the chassis only to return back to the MT to pick up the wheeled import.

1.3 Centralized Processing of Chassis

This dissertation examines the concept of the “Centralized Processing of Chassis.” The concept revolves around an off-dock terminal (or several off-dock terminals), referred to as CPFs. A CPF is located close to the port, where trucks will go to exchange chassis, thereby reducing traffic at the MTs and resulting in reduced travel times and reduced congestion. The effort is divided into two major studies: one at the strategic planning level, and one at the operational level for individual trucking companies.

In the first study presented in Section 2, an analytical framework for modeling and optimization of chassis movements in transportation networks with CPFs is developed. The developed methodology is then used to study and analyze the optimal CPF locations and to evaluate the potential benefits of the Centralized Processing of Chassis concept. The analytical models and optimization are defined for export and import transactions, and take into account real-world financial constraints by evaluating the optimal CPF quantity and locations as a function of the policy makers’ available budget.

In the second study described in Section 3, scheduling of chassis and container movements is optimized at the operational level for individual trucking companies, when CPFs are available for use within a major metropolitan area. Time-varying dynamic models for the movements of chassis and containers are developed and used in the optimization process. The effectiveness of the developed methodology is evaluated through simulation scenarios.

In both cases the Long Beach/Los Angeles (LB/LA) port area is leveraged as a particular case study of interest. However, the methodology in general would be applicable in most port drayage scenarios. The methodologies developed herein could contribute to improving the traffic conditions in the areas
surrounding the ports by modifying the patterns of truck trips to the ports. Hence, they have the potential to reduce traffic congestion, air pollution and the economic loss that would result from unnecessary delays and truck waiting times.
2 CHASSIS PROCESSING LOCATION OPTIMIZATION

The main objective of this study is to analyze and develop an analytical framework for modeling and optimization of the concept of Centralized Processing of Chassis around marine container terminals. A detailed example of the application of the developed methodology is presented through a case study, which focuses on the POLB and POLA, and on the areas in the vicinity of the ports. The case study considers the locations of all existing container MTs in the POLB and POLA complex, the locations of a number of existing trucking companies in the greater LB/LA geographical area, and a set of potential locations for CPFs. The developed methodology is used to study and analyze the optimal CPF locations, and to evaluate the potential benefits of the Centralized Processing of Chassis concept. The analytical models and optimization are first defined for import only transactions (requiring only chassis retrieval at CPF locations) under the assumption that any of the potential CPF locations are available and can be used. This model is then expanded to export and import transactions, using only a small number of CPFs, which have storage capacity limitations. The expanded model, which optimizes the chassis exchange process with a smaller number of CPFs allows for policy makers to make decisions for optimal solutions while taking into account budgetary constraints.

The methodologies developed herein could contribute to improving the traffic conditions in the areas surrounding the ports, by modifying the patterns of truck trips to the ports. Analysis based on the simulation results, shows that the total travel time can be improved up to 20% when using CPFs to store and retrieve chassis as compared to the baseline situation where the chassis are retrieved directly from the MTs. In addition, optimal locations and CPF combinations are recommended, and a sensitivity analysis is performed. The sensitivity analysis assesses the robustness of the optimal solution by evaluating how it varies as a function of the CPF capacities, the total numbers of transactions, and the ratio of export to import transactions.

The general concept for the centralized processing of chassis is captured in Figure 6. For the purposes of this study the origin points are generically referred to as TCs (trucking companies) and the destination
points are generically referred to as MTs (marine terminals). I assume that there exist $J$ regional TCs which can use any of $K$ potential CPFs as they perform various transactions with the $L$ local MTs. It is assumed that the travel time between each of the possible locations is also included, where the travel time (cost) between the $j^{th}$ TC, $TC_j$, and $k^{th}$ CPF, $CPF_k$, is given by $C_{TC_jCPF_k}$, the travel time between $CPF_k$ and the $l^{th}$ MT, $MT_l$, is given by $C_{CPF_kMT_l}$, and the travel time between $TC_j$ and $MT_l$ is given by $C_{TC_jMT_l}$.

![Diagram]

**Figure 6. Schematic of Centralized Processing of Chassis concept.**

### 2.1 Analytical Models and Optimization

In this section, a general analytical framework for the Centralized Processing of Chassis concept is developed. Using this framework, the optimal number and optimal locations of chassis processing facilities may be identified.
Due to the recent changes in chassis leasing policies, such as the introduction of the grey chassis pool in the POLB and POLA, it is assumed in this analysis that chassis of similar types are interchangeable and transactions do not need to take into account chassis ownership.

Given:

- the locations and storage capacities of the destination points, $MT_l, l = 1, \ldots, L$
- the locations of the origin points $TC_j, j = 1, \ldots, J$
- a set of transactions needed between $TC_j, j = 1, \ldots, J$, and $MT_l, l = 1, \ldots, L$, during a time interval of interest
- the locations of potential $CPF_k$ sites, $k = 1, \ldots, K$

the objective herein is:

- to minimize the total travel time during the time interval of interest for all transactions by allocating one or more sites (among all potential sites) for CPFs, and determining the optimal routing of each of the transactions through CPFs.

### 2.1.1 Import Only with Unlimited Chassis Processing Facility Usage

For the analytical formulation of the Centralized Processing of Chassis concept, the following notation is used.

- $J$ Number of TCs collaborating with the CPFs
- $K$ Number of potential sites for CPFs
- $L$ Number of participating MTs
- $TC_j$ The $j^{th}$ TC (origin point) $j \in \{1, \ldots, J\}$
- $CPF_k$ The $k^{th}$ CPF $k \in \{1, \ldots, K\}$
- $MT_l$ The $l^{th}$ MT $l \in \{1, \ldots, L\}$
- $C_{TC_jF_k}$ Cost of transactions between $TC_j$ and $CPF_k$
$C_{F_k M_l}$ Cost of transactions between $CPF_k$ and $MT_l$

$C_{T_j M_l}$ Cost of transactions between $TC_j$ and $MT_l$

$n_{jl}$ Number of Type 2 (import) transactions from $MT_l$ to $TC_j$

$U_k$ Storage capacity of chassis at $CPF_k$

$x_{jkl}$ Number of Type 2 (import) transactions from $MT_l$ to $TC_j$ routed through $CPF_k$

$y_{jl}$ Number of Type 2 (import) transactions routed directly from $MT_l$ to $TC_j$

Note that in this model some assumptions have been made in order to simplify the modeling process: The cost of all trips between TCs to CPFs and to MTs has been assumed to be symmetric, i.e. the cost is the same whether the trip is from a TC to CPF or from a CPF to TC, and similarly for MTs. The cost is assumed fixed without any time-varying component that may result from traffic variations at different times of the day. No specific distinction is given, in this problem, as to alternate chassis / container sizes or the complete set of transaction types defined above. Initially, I assume only Type 2 transactions (grounded imports) using a single common chassis.

These simplifications allow the problem to be defined as an integer linear program where the objective function is given by:

$$\begin{align*}
\text{min} & \quad \sum_{j=1}^{J} \sum_{k=1}^{K} C_{TjF_k} \left( \sum_{l=1}^{L} x_{jkl} \right) + \sum_{k=1}^{K} \sum_{l=1}^{L} C_{F_k M_l} \left( \sum_{j=1}^{J} x_{jkl} \right) + \sum_{j=1}^{J} \sum_{l=1}^{L} C_{TjM_l} y_{jl} \\
\text{s.t.} & \quad \sum_{k=1}^{K} x_{jkl} + y_{jl} = n_{jl} \\
& \quad \sum_{j=1}^{J} \sum_{l=1}^{L} x_{jkl} \leq U_k \\
& \quad x_{jkl} \in \mathbb{N}^0 \\
& \quad y_{jl} \in \mathbb{N}^0
\end{align*}$$

(1)

(2)

(3)

(4)

(5)
In Equation (1) the total travel time for completion of all transactions is minimized. Decision variables are $x_{jkl}$ (number of transactions from $TC_j$ to $MT_l$ routed through $CPF_k$) and $y_{jl}$ (number of transactions routed directly from $TC_j$ to $MT_l$). Constraint (2) is the conservation of transactions constraint, forcing the total number of transactions routed directly to MTs and through CPFs to be equal to the total transaction demand. Constraint (3) ensures that the CPF chassis supply is sufficient to meet import demands. Constraints (4)-(5) ensure that the number of transactions routed on any possible path are non-negative integers (i.e. belong to the set of natural numbers including zero).

In order to use a standard solver this was reformulated as an integer linear program below where the objective function is then given by

$$\min \quad f^T x$$

subject to

$$x \in \mathbb{Z}$$

$$Ax \leq \alpha$$

$$Bx = \beta$$

$$x \geq \gamma$$

where

$$x = \begin{bmatrix}
  x_{1,1,1} \\
  \vdots \\
  x_{1,1,L} \\
  \vdots \\
  x_{1,K,1} \\
  \vdots \\
  x_{1,K,L} \\
  x_{2,1,1} \\
  \vdots \\
  x_{J,K,L} \\
  y_{1,1} \\
  \vdots \\
  y_{J,L} \\
\end{bmatrix}$$

$$f = \begin{bmatrix}
  C_{T_1F_1} + C_{F_1M_1} \\
  \vdots \\
  C_{T_1F_1} + C_{F_1M_L} \\
  \vdots \\
  C_{T_1F_K} + C_{F_KM_1} \\
  \vdots \\
  C_{T_1F_K} + C_{F_KM_L} \\
  C_{T_2F_1} + C_{F_1M_1} \\
  \vdots \\
  C_{T_2F_1} + C_{F_1M_L} \\
  \vdots \\
  \vdots \\
  \vdots \\
  C_{T_JF_K} + C_{F_KM_1} \\
  \vdots \\
  C_{T_JF_K} + C_{F_KM_L} \\
  C_{T_1M_1} \\
  \vdots \\
  C_{T_1M_2} \\
  \vdots \\
  C_{T_JM_L}
\end{bmatrix}$$

(11)
\[
A = \begin{bmatrix}
\text{rep}_j([1^T_L, 0^T_{(K-1)+L}], 0^T_{j+L}) \\
\text{rep}_j([0^T_L, 1^T_L, 0^T_{(K-2)+L}], 0^T_{j+L}) \\
\vdots \\
\text{rep}_j([0^T_{(K-1)+L}, 1^T_L, 0^T_{(K-k)+L}], 0^T_{j+L}) \\
\text{rep}_j([0^T_{(K-1)+L}, 1^T_L])
\end{bmatrix}
\quad \alpha = \begin{bmatrix}
U_1 \\
U_2 \\
\vdots \\
U_K
\end{bmatrix}
\]

\[
B = \begin{bmatrix}
\text{rep}_K((e_1^L)^T), 0^T_{(J-1)+K+L}, (e_1^{j+L})^T \\
\text{rep}_K((e_2^L)^T), 0^T_{(J-1)+K+L}, (e_2^{j+L})^T \\
\vdots \\
0^T_{(J-1)+K+L}, \text{rep}_K((e_1^L)^T), 0^T_{(J-1)+K+L}, (e_1^{j+L})^T \\
0^T_{(J-1)+K+L}, \text{rep}_K((e_2^L)^T), (e_2^{j+L})^T \\
\vdots \\
0^T_{(J-1)+K+L}, \text{rep}_K((e_1^L)^T), (e_1^{j+L})^T \\
0^T_{(J-1)+K+L}, \text{rep}_K((e_2^L)^T), (e_2^{j+L})^T
\end{bmatrix}
\quad \beta = \begin{bmatrix}
n_{1,1} \\
n_{1,2} \\
\vdots \\
n_{1,L} \\
n_{2,1} \\
n_{2,2} \\
\vdots \\
n_{2,L} \\
\vdots \\
n_{J,1} \\
n_{J,2} \\
\vdots \\
n_{J,L}
\end{bmatrix}
\]

\[
\gamma = [0^T_{(J-K+L)+L}]^T
\]

where
\[
e_{a}^b = [e_1, e_2, ..., e_b]^T, e_a = 1, e_j = 0 \forall j \neq a
\]

\[
\text{rep}_b(a) = [a_1, a_2, ..., a_b], a_j = a \forall j
\]

\[
0_b = [\alpha_1, \alpha_2, ..., \alpha_b]^T, \alpha_j = 0 \forall j
\]

\[
1_b = [\alpha_1, \alpha_2, ..., \alpha_b]^T, \alpha_j = 1 \forall j
\]

### 2.1.2 Import Only with Limited Chassis Processing Facility Usage

The mathematical formulation in the section above assumed that an unlimited number of CPFs could be used from within the potential set of CPFs identified. This may be unrealistic as cost constraints of establishing and maintaining each CPF may prevent all sites from being used. To address this problem I put a constraint on the cost by setting a limit on the total number of CPFs which can be used. This requires two additional variables listed below.

\[
N_{\text{max}} \quad \text{Maximum number of allowable CPFs}
\]

\[
z_k \quad \text{Variable identifying if CPF k has any transactions}
\]
where

\[
z_k = \begin{cases} 
1, & \sum_{j=1}^{J} L_{l} \sum_{l=1}^{L} (x_{jkl}) \geq 0 \\
0, & \text{otherwise} 
\end{cases} \quad k = 1, \ldots, K \tag{19}
\]

\[
\sum_{k=1}^{K} z_k \leq N_{\text{max}} \tag{20}
\]

This can be updated further to take into account the potential varying cost of establishing and maintaining the CPFs between each potential site.

Assuming these costs are known as well as the maximum allowable cost for establishing and maintaining the CPFs, Equation (20) would be replaced with Equation (21) below.

\[
\sum_{k=1}^{K} z_k * C_k \leq C_{\text{max}} \tag{21}
\]

where

\[
C_{\text{max}} \quad \text{Maximum cost allowable for establishing and maintaining CPFs}
\]

\[
C_k \quad \text{Cost of establishing and maintaining CPF k}
\]

Putting this altogether, this can be written as an integer program as follows.

\[
\begin{align*}
\text{min} & \quad \sum_{j=1}^{J} \sum_{k=1}^{K} C_{T,jF_k} \left( \sum_{l=1}^{L} x_{jkl} \right) + \sum_{k=1}^{K} \sum_{l=1}^{L} C_{F,kM_l} \left( \sum_{j=1}^{J} x_{jkl} \right) + \sum_{j=1}^{J} \sum_{l=1}^{L} C_{T,jM_l} y_{ji} \\
\text{s.t.} & \quad \sum_{k=1}^{K} x_{jkl} z_k + y_{ji} = n_{jl} \quad j = 1, \ldots, J; \ l = 1, \ldots, L
\end{align*} \tag{22}
\]
\[
\sum_{j=1}^{J} \sum_{l=1}^{L} x_{jkl} z_k \leq U_k \quad k = 1, ..., K
\]
(24)

\[
\sum_{k=1}^{K} z_k \cdot C_k \leq C_{\text{max}}
\]
(25)

\[
x_{jkl} \in \mathbb{N}^0 \quad j = 1, ..., J; \; l = 1, ..., L; \; k = 1, ..., K
\]
(26)

\[
y_{jl} \in \mathbb{N}^0 \quad j = 1, ..., J; \; l = 1, ..., L
\]
(27)

\[
z_k \in \{0, 1\} \quad k = 1, ..., K
\]
(28)

Note that Equation (22), (25), (26), and (27) are identical to Equations (1), (21), (4), and (5) respectively and have only been presented here again out of convenience. Constraints of Equations (23), (24), and (28) have been included to combine the constraints of Equation (19) with those of Equations (2) and (3) as a quadratic linear integer program.

### 2.1.3 Export and import Formulation

In this section the import only formulation is expanded to include both export and import transactions (i.e. Type 1 and Type 2 transactions). The notation used is as follows:

\[
m_{jl} \quad \text{Number of Type 1 (export) transactions from } TC_j \text{ to } MT_l
\]

\[
U_{k,0} \quad \text{Initial Storage capacity of chassis at } CPF_k
\]

\[
\alpha_{jkl} \quad \text{Number of Type 1 (export) transactions from } TC_j \text{ to } MT_l \text{ routed through } CPF_k
\]

\[
\beta_{jl} \quad \text{Number of Type 1 (export) transactions routed directly from } TC_j \text{ to } MT_l
\]
With the addition of the Type 1 transactions (grounded exports) the updated integer linear program can be written as follows.

\[
\min \sum_{j=1}^{J} \sum_{k=1}^{K} C_{T_jF_k} \left( \sum_{l=1}^{L} x_{jkl} \right) + \sum_{k=1}^{K} \sum_{l=1}^{L} C_{F_kM_l} \left( \sum_{j=1}^{J} x_{jkl} \right) + \sum_{j=1}^{J} \sum_{l=1}^{L} C_{T_jM_l} y_{jl} \\
+ \sum_{j=1}^{J} \sum_{k=1}^{K} C_{T_jF_k} \left( \sum_{l=1}^{L} \alpha_{jkl} \right) + \sum_{k=1}^{K} \sum_{l=1}^{L} C_{F_kM_l} \left( \sum_{j=1}^{J} \alpha_{jkl} \right) \\
+ \sum_{j=1}^{J} \sum_{l=1}^{L} C_{T_jM_l} \beta_{jl}
\]

s.t. \[
\sum_{k=1}^{K} x_{jkl} + y_{jl} = n_{jl} \quad j = 1, \ldots, J; \ l = 1, \ldots, L \quad (30)\\
\sum_{k=1}^{K} \alpha_{jkl} + \beta_{jl} = m_{jl} \quad j = 1, \ldots, J; \ l = 1, \ldots, L \quad (31)\\
\sum_{j=1}^{J} \sum_{l=1}^{L} (x_{jkl} - \alpha_{jkl}) \leq U_{k,0} \quad k = 1, \ldots, K \quad (32)\\
\sum_{j=1}^{J} \sum_{l=1}^{L} (x_{jkl} - \alpha_{jkl}) \geq U_{k,0} - U_k \quad k = 1, \ldots, K \quad (33)\\
x_{jkl} \in \mathbb{N}^{0} \quad j = 1, \ldots, J; \ l = 1, \ldots, L; \ k = 1, \ldots, K \quad (34)\\
y_{jl} \in \mathbb{N}^{0} \quad j = 1, \ldots, J; \ l = 1, \ldots, L \quad (35)\\
\alpha_{jkl} \in \mathbb{N}^{0} \quad j = 1, \ldots, J; \ l = 1, \ldots, L; \ k = 1, \ldots, K \quad (36)\\
\beta_{jl} \in \mathbb{N}^{0} \quad j = 1, \ldots, J; \ l = 1, \ldots, L \quad (37)
\]

In Equation (29) the total travel time for completion of all transactions is minimized using the decision variables \(x_{jkl}, y_{jl}, \alpha_{jkl}, \text{ and } \beta_{jl}\). Constraints (30) and (31) provide the equality constraint forcing the total...
number of export and import transactions routed directly to/from MTs and routed through CPFs to be equal to the total transaction demand. Constraints (32) and (33) ensure that the chassis supply and capacity at each CPF is sufficient to meet export and import demands. Constraints (34) through (37) ensure that the number of transactions routed on any possible path are non-negative integers (i.e. belong to the set of natural numbers including zero).

In order to use a standard solver this was once again reformulated as an integer linear program in accordance with Equations (6) through (10), such that

\[
\begin{align*}
    x & = \begin{bmatrix}
        x_{\text{orig}} \\
        \alpha_{1,1,1} \\
        \vdots \\
        \alpha_{1,1,L} \\
        \vdots \\
        \alpha_{1,K,1} \\
        \vdots \\
        \alpha_{1,K,L} \\
        \alpha_{2,1,1} \\
        \vdots \\
        \alpha_{J,K,1} \\
        \beta_{1,1} \\
        \vdots \\
        \beta_{1,2} \\
        \vdots \\
        \beta_{J,L}
    \end{bmatrix} \\
    f & = \begin{bmatrix}
        f_{\text{orig}} \\
        C_{T_1F_1} + C_{F_1M_1} \\
        \vdots \\
        C_{T_1F_1} + C_{F_1M_L} \\
        \vdots \\
        C_{T_1F_K} + C_{F_KM_1} \\
        \vdots \\
        C_{T_1F_K} + C_{F_KM_L} \\
        C_{T_2F_1} + C_{F_1M_1} \\
        \vdots \\
        C_{T_2F_1} + C_{F_1M_L} \\
        \vdots \\
        C_{T_JF_K} + C_{F_KM_1} \\
        C_{T_JF_K} + C_{F_KM_L} \\
        C_{T_1M_1} \\
        \vdots \\
        C_{T_1M_2} \\
        \vdots \\
        C_{T_JM_1} \\
        C_{T_JM_L}
    \end{bmatrix}
\end{align*}
\]
\[ A = \begin{bmatrix}
    \text{rep}_j\left(\begin{bmatrix}1_L^T, 0_{(K-1)+L}^T\end{bmatrix}\right), 0_{j+L}^T, -\text{rep}_j\left(\begin{bmatrix}1_L^T, 0_{(K-1)+L}^T\end{bmatrix}\right), 0_{j+L}^T \\
    \text{rep}_j\left(\begin{bmatrix}0_L^T, 1_L^T, 0_{(K-2)+L}^T\end{bmatrix}\right), 0_{j+L}^T, -\text{rep}_j\left(\begin{bmatrix}0_L^T, 1_L^T, 0_{(K-2)+L}^T\end{bmatrix}\right), 0_{j+L}^T \\
    \vdots \\
    \text{rep}_j\left(\begin{bmatrix}0_{(K-1)+L}^T, 1_L^T, 0_{(K-2)+L}^T\end{bmatrix}\right), 0_{j+L}^T, -\text{rep}_j\left(\begin{bmatrix}0_{(K-1)+L}^T, 1_L^T, 0_{(K-2)+L}^T\end{bmatrix}\right), 0_{j+L}^T \\
    \text{rep}_j\left(\begin{bmatrix}0_{(K-1)+L}^T, 1_L^T, 0_{(K-2)+L}^T\end{bmatrix}\right), 0_{j+L}^T, -\text{rep}_j\left(\begin{bmatrix}0_{(K-1)+L}^T, 1_L^T, 0_{(K-2)+L}^T\end{bmatrix}\right), 0_{j+L}^T \\
    \vdots \\
    \text{rep}_j\left(\begin{bmatrix}0_{(K-1)+L}^T, 1_L^T, 0_{(K-2)+L}^T\end{bmatrix}\right), 0_{j+L}^T, -\text{rep}_j\left(\begin{bmatrix}0_{(K-1)+L}^T, 1_L^T, 0_{(K-2)+L}^T\end{bmatrix}\right), 0_{j+L}^T
  \end{bmatrix}
\]

\[ B = \begin{bmatrix}
    \text{rep}_j\left(\begin{bmatrix}e_1^T\end{bmatrix}\right), 0_{(j-1)+K*L,L}^T, (e_1^{j+L})^T, 0_{j+K*L+J*L}^T \\
    \text{rep}_j\left(\begin{bmatrix}e_2^T\end{bmatrix}\right), 0_{(j-1)+K*L,L}^T, (e_2^{j+L})^T, 0_{j+K*L+J*L}^T \\
    \vdots \\
    0_{(j-1)+K*L,L}^T, \text{rep}_j\left(\begin{bmatrix}e_1^T\end{bmatrix}\right), 0_{(j-1)+K*L,L}^T, (e_{j-1}^{j+L})^T, 0_{j+K*L+J*L}^T \\
    0_{(j-1)+K*L,L}^T, \text{rep}_j\left(\begin{bmatrix}e_2^T\end{bmatrix}\right), 0_{(j-1)+K*L,L}^T, (e_{j-1}^{j+L})^T \\
    \vdots \\
    0_{j+K*L+J*L}^T, \text{rep}_j\left(\begin{bmatrix}e_1^T\end{bmatrix}\right), 0_{j+K*L+J*L}^T, (e_{j-1}^{j+L})^T \\
    0_{j+K*L+J*L}^T, \text{rep}_j\left(\begin{bmatrix}e_2^T\end{bmatrix}\right), 0_{j+K*L+J*L}^T
  \end{bmatrix}
\]

\[ \alpha = \begin{bmatrix}
    U_{1,0} \\
    U_{2,0} \\
    \vdots \\
    U_{K,0} \\
    U_1 - U_{1,0} \\
    U_2 - U_{2,0} \\
    \vdots \\
    U_K - U_{K,0}
  \end{bmatrix}
\]

(39)
\[ \beta = \begin{bmatrix} \beta_{\text{orig}} \\ m_{1,1} \\ \vdots \\ m_{1,L} \\ m_{2,1} \\ \vdots \\ m_{2,L} \\ \vdots \\ m_{J,L} \end{bmatrix} \]

\[ \gamma = \left[ 0_{2\times(J+K+L+L)} \right]^T \]

where

\[ e_a^b = [e_1, e_2, \ldots, e_b]^T, e_a = 1, e_j = 0 \forall j \neq a \]

\[ \text{rep}_b(a) = [a_1, a_2, \ldots, a_b], a_j = a \forall j \]

\[ 0_b = [\alpha_1, \alpha_2, \ldots, \alpha_b]^T, \alpha_j = 0 \forall j \]

\[ 1_b = [\alpha_1, \alpha_2, \ldots, \alpha_b]^T, \alpha_j = 1 \forall j \]

Finally, in order to update for a limited supply of chassis, Equation (19) can be updated as

\[ z_k = \begin{cases} 1, & \sum_{j=1}^{J} \sum_{l=1}^{L} (x_{jkl} + \alpha_{jkl}) \geq 0 \\ 0, & \text{otherwise} \end{cases} \]

The general optimization formulations described above have been applied to the analysis of a specific case study created for the POLB and POLA. The following sections describe the set-up of the case study and present results based on a variety of simulation scenarios and conditions.

### 2.2 Case Study Model Implementation

A case study using the optimization formulation described in Section 2.1 was performed for the POLB and POLA. The case study uses the MTs in the POLB and POLA complex, and a number of TCs and
potential CPF locations in the vicinity of the ports. The selection methods for the TCs and CPF locations are described in greater detail in the following sections.

A general overview of the local LB/LA port area is shown in Figure 7, which indicates:

- The MT locations (destination points) at the POLB and POLA. The MTs are shown as the color-coded areas on the map.
- The TCs (points of origin) used for the case study. The TCs are distributed in a wide area around the ports, and are shown as yellow dots on the map.
- The potential CPF locations used in the case study, which are distributed in a wide area around the ports and are shown as white pins on the map.

The optimization formulations described in Section 2.1 were applied in the case study to evaluate the potential CPF locations. The total travel time for all trucks within a time period of interest is minimized, for an estimated total number of transactions based upon historical data, using estimates of CPF capacities, and estimated travel times between locations queried from the Google Distance Matrix (GDM) © Application Program Interface (API).
2.2.1 Marine Terminals

The POLB and POLA have terminals which cover various categories of exports and imports such as automotive, dry bulk, break bulk liquid and containers. This study concentrates on export/import of grounded containers and efficient retrieval and use of their associated chassis by truck. The terminals at the POLB and POLA which handle containers in this manner are listed in Table 1 and shown in Figure 8.
Table 1. Locations of POLB and POLA terminals used in the case study.

<table>
<thead>
<tr>
<th>MT ID</th>
<th>Name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ITS (K-Line)</td>
<td>Pier G E, Long Beach, CA 90802, USA</td>
</tr>
<tr>
<td>2</td>
<td>LBCT (OOCL)</td>
<td>Pier F Ave, Long Beach, CA 90802, USA</td>
</tr>
<tr>
<td>3</td>
<td>Pacific Container Terminal (COSCO)</td>
<td>Harbor Scenic Way, Long Beach, CA 90802, USA</td>
</tr>
<tr>
<td>4</td>
<td>SSA - Pier A</td>
<td>Pier C St, Long Beach, CA 90802, USA</td>
</tr>
<tr>
<td>5</td>
<td>SSA (MSC, Zim, SMA/CGM)</td>
<td>Pier A Way, Long Beach, CA 90802, USA</td>
</tr>
<tr>
<td>6</td>
<td>TTI (Hanjin)</td>
<td>Hanjin Rd, Long Beach, CA 90802, USA</td>
</tr>
<tr>
<td>7</td>
<td>APM Terminals Pacific</td>
<td>Navy Way Terminal Island, CA 90731</td>
</tr>
<tr>
<td>8</td>
<td>California United Terminals</td>
<td>Navy Way, Terminal Island, CA 90731</td>
</tr>
<tr>
<td>9</td>
<td>China Shipping North America</td>
<td>John S. Gibson Boulevard San Pedro, CA 90731</td>
</tr>
<tr>
<td>10</td>
<td>Eagle Marine Services</td>
<td>Terminal Way, Los Angeles, CA 90731</td>
</tr>
<tr>
<td>11</td>
<td>Everport Terminal Services</td>
<td>Terminal Island Way Terminal Island, CA 90731</td>
</tr>
<tr>
<td>12</td>
<td>TraPac, Inc</td>
<td>South Neptune Avenue, Wilmington, CA 90744</td>
</tr>
<tr>
<td>13</td>
<td>Yang Ming Marine Transport</td>
<td>John S. Gibson Boulevard, San Pedro, CA 90731</td>
</tr>
<tr>
<td>14</td>
<td>Yusen Terminal (Nyk Yusen)</td>
<td>New Dock Street Terminal Island, CA 90731</td>
</tr>
</tbody>
</table>
Figure 8. POLB & POLA terminal locations.

Loaded outbound (export) and inbound (import) quantities through the POLB and POLA for 2017 are included in Table 2 (Port of Los Angeles Statistics, 2018), (Port of Long Beach Statistics, 2018).

Table 2. POLB and POLA export and import statistics for 2017.

<table>
<thead>
<tr>
<th></th>
<th>loaded export</th>
<th>loaded import</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEU POLB</td>
<td>1,470,514</td>
<td>3,863,187</td>
</tr>
<tr>
<td>TEU POLA</td>
<td>1,899,934</td>
<td>4,716,089</td>
</tr>
<tr>
<td>TEU Total (Year)</td>
<td>3,370,448</td>
<td>8,579,276</td>
</tr>
<tr>
<td>TEU Total Avg (Day)</td>
<td>9,234</td>
<td>23,505</td>
</tr>
</tbody>
</table>
2.2.2 Trucking Companies

A representative sample of seventy-one TCs which service the POLB and POLA was used in this case study. In order to select this sample, an initial list of TCs was created from an internet drayage directory which includes all companies operating within Los Angeles County. Since the location of the TCs is a critical variable for the optimization problem, any company whose address was not included in the drayage directory was eliminated from the list. The final list contains all companies with known addresses using chassis. In the analysis herein the number of daily transactions between MTs and TCs was assumed to be a fixed value between each TC and each MT. In the initial analysis, the number of total daily forty-foot import transactions was set at 50,000 based on forecasts of total daily port trips (Weikel, 2015). The sensitivity analysis used 5,000 export and 10,000 import daily forty-foot transactions based on the average daily export and import container traffic noted in Table 2.

2.2.3 Central Processing Facilities

Sixteen potential CPF locations were identified by searching for vacant land within a fifteen-mile radius of the POLB and POLA as seen in Figure 7. The capacities of these locations were estimated by using the Google Earth © polygon built-in feature to calculate an approximate square footage. Several CPF layout options and chassis stacking methodologies were evaluated and a horizontal storage layout with a maximum of three chassis stacked on top of each other was selected for the case study. Using the estimated square footage of the potential CPF location, the number of forty-foot chassis which could fit in that area was determined assuming allocations for access roads, two blocks of stacked chassis (each stacked three high), and another smaller block of un-stacked chassis for ease of access in order to minimize chassis retrieval times. An example of the layout for a 5,000 x 5,000 foot area is included in Figure 9 below. For this example, the maximum number of forty-foot chassis which could be stored in this area was estimated at 170,000.
Figure 9. Example of chassis stacking layout within a CPF.

The locations and capacity estimates for each of the CPFs included in the case study are provided in Table 3. The table shows the street name and zip code of the potential CPF locations, but the street address numbers have been removed.
Table 3. Potential CPF locations and capacities for chassis storage.

<table>
<thead>
<tr>
<th>CPF ID</th>
<th>Address</th>
<th>Estimated Capacity (chassis units)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Golden Ave, Long Beach, CA 90806, USA</td>
<td>7,467</td>
</tr>
<tr>
<td>2</td>
<td>Via Oro Ave, Long Beach, CA 90810, USA</td>
<td>4,048</td>
</tr>
<tr>
<td>3</td>
<td>River Ave, Long Beach, CA 90810, USA</td>
<td>16,779</td>
</tr>
<tr>
<td>4</td>
<td>E 213th St, Carson, CA 90746, USA</td>
<td>6,350</td>
</tr>
<tr>
<td>5</td>
<td>E Del Amo Blvd, Carson, CA 90746, USA</td>
<td>20,551</td>
</tr>
<tr>
<td>6</td>
<td>Long Beach Blvd, Long Beach, CA 90805, USA</td>
<td>20,469</td>
</tr>
<tr>
<td>7</td>
<td>Long Beach Blvd, Long Beach, CA 90805, USA</td>
<td>557</td>
</tr>
<tr>
<td>8</td>
<td>S Sportsman Dr, Compton, CA 90221, USA</td>
<td>2,161</td>
</tr>
<tr>
<td>9</td>
<td>Atlantic Ave, Long Beach, CA 90805, USA</td>
<td>757</td>
</tr>
<tr>
<td>10</td>
<td>Alondra Blvd, Paramount, CA 90723, USA</td>
<td>2,084</td>
</tr>
<tr>
<td>11</td>
<td>Alondra Blvd, Paramount, CA 90723, USA</td>
<td>293</td>
</tr>
<tr>
<td>12</td>
<td>Torrance Blvd, Carson, CA 90745, USA</td>
<td>3,192</td>
</tr>
<tr>
<td>13</td>
<td>W Del Amo Blvd, Torrance, CA 90502, USA</td>
<td>1,120</td>
</tr>
<tr>
<td>14</td>
<td>W Del Amo Blvd, Torrance, CA 90502, USA</td>
<td>1,877</td>
</tr>
<tr>
<td>15</td>
<td>S Figueroa St, Wilmington, CA 90744, USA</td>
<td>2,258</td>
</tr>
<tr>
<td>16</td>
<td>Lomita Blvd, Carson, CA 90745, USA</td>
<td>6,959</td>
</tr>
</tbody>
</table>

2.2.4 Travel Time Between Locations

The travel times between all TCs, CPFs and MTs were calculated using the GDM API. The addresses of each location were input into the GDM API which in turn provided the non-traffic (ideal) travel times between all given locations. The travel times were used to populate the three matrices identified in Section 2.1, including the cost (travel time) of transactions between TCs and CPFs (consisting of 1,136 individual elements $C_{TF_k}$), the cost of transactions between CPFs and MTs (consisting of 224 individual elements $C_{FM_l}$), and the cost of transaction between TCs and MTs (consisting of 994 individual elements $C_{TM_l}$). In the results provided herein it was assumed that for any locations $x$ and $y$, the time to travel from location $x$ to location $y$ ($C_{xy}$) was equal to that to travel from location $y$ to location $x$ ($C_{yx}$), i.e.

$$C_{xy} = C_{yx} \quad \forall x, y$$ (49)
2.2.5 Improved Chassis Processing Times at CPFs

A key assumption in the potential for improvement when using CPFs rather than direct routing to the MTs is that processing of chassis at the CPFs would be more efficient than that at the MTs. To illustrate the potential for improved processing times at CPFs, the chassis pick-up process is shown in Figure 10 below. In general, the average time for a driver to obtain a chassis is approximately twelve minutes; however it can vary greatly, where if an issue arises in the checkout process which the driver cannot personally address, an average of an additional hour may be required (The Tioga Group, Research, & Engineering, 2011).

![Chassis Pick-up Process Diagram](image)

**Figure 10. Chassis pick-up process.**

At the CPFs, it is assumed that there will be an opportunity to prescreen chassis and limit truck trips to the roadability canopy, such that the MTs would, on average, require additional processing time for any chassis interactions as compared to the CPFs. This “Additional Processing Time” at the MTs is denoted by $P$, and provides a measure of the relative advantage of routing a transaction through a CPF over routing the transaction directly to the MT. The parameter $P$ is defined as the difference between the
average time it takes to retrieve a chassis stored in a MT and the average time it takes to retrieve a chassis stored in a CPF:

\[ P = T_M - T_F \text{ (seconds)} \]  

where

\( T_M \) is the average chassis retrieval time at a MT, in seconds

and \( T_F \) is the average chassis retrieval time at a CPF, in seconds

In the following section, results of the simulation scenarios are calculated and plotted against the additional processing time (parameter \( P \)). This provides a range for the predicted benefit of using the CPFs as a function of the realized increase in efficiency when routing chassis through CPFs. Values for \( P \) up to twenty minutes were included in this study.

### 2.3 Case Study Simulation Results

This section presents the results of several simulation scenarios for the case study, based on the optimization formulation described previously. In order to validate the results and assess the developed methodology, the simulations progress from simpler scenarios on a reduced-node model to the more complicated scenarios applied to the full model.

#### 2.3.1 Optimization Software

In order to implement the integer linear program equations from Section 2.1, the equations were first reformulated in order to use a standard solver as noted in Sections 2.1.1 and 2.1.3 for the import only and export/import formulations respectively. The reformulated equations were then coded in Matlab version R2016a, and the problem was solved using the intlinprog function and the following configuration:
• **Integer Tolerance:** The tolerance for a number to be considered an integer was $1e^{-5}$.

• **Linear Program Solution (without linear constraints):** First the linear program was solved without the linear constraints, which included preprocessing to reduce the problem size, per (Andersen, 1995) (Mészáros C., 2003), followed by implementation of a Dual Simplex algorithm resulting in a root solution / lower bound for the objective function.

• **Cut Generation:** Integer preprocessing, per (Savelsbergh, 1994), was performed, followed by sequencing through Mixed-integer rounding, Gomory, Clique, Cover and Flow cover cuts to add additional linear constraints to the problem to restrict the solutions and force them to be closer to integers. Ten passes through the five cut generation methods noted above were performed before entering the branch-and-bound phase.

• **Branch and Bound:** Rounding and hybrid neighborhood search / local branching heuristic search approaches were used to find feasible points to provide an upper bound on the objective function (Danna, 2005) (Mathworks Support: Mixed-Integer Linear Programming Algorithms, 2018) prior to the branch and bound process. The branch and bound process then was performed with the fractional variable for splitting selected as that with the maximum pseudocost, see (Mathworks Support: Mixed-Integer Linear Programming Algorithms, 2018), which would have resulted in the largest increase in the lower bound. The nodes to explore after branching were then selected based upon the “best projection” criterion. The branch and bound process was then terminated when either the relative lower and upper bound difference in the objective function became less than 0.0001 (which equates to stopping when bounds are within approximately 0.01% of each other), or the number of nodes explored in the branch and bound process reached the maximum value of 10,000,000.
For the network sizes and CPF quantities implemented herein, the quadratic integer program equations were able to be implemented through recursive calls to the same integer linear program over the complete set of potential CPF permutations.

2.3.2 Import Only with Unlimited CPF Usage

The case study was first performed using the import only integer linear mathematical formulation with unlimited chassis processing facility usage described in Section 2.1.1.

2.3.2.1 Reduced-Node Model Evaluation: Six Nodes

As an initial assessment of the linear program formulation, a reduced-node model was generated utilizing subsets of the TCs, MTs and CPFs. Two nodes from each set of TCs, MTs and CPFs were selected, so that the model used for the initial assessment of the methodology contains a total of six nodes. The selected locations and travel times between them are included in Table 4 through Table 6. Figure 11 shows the complete schematic for the reduced-node problem.

For this simplified scenario, it is assumed that 12,500 import transactions are required to be completed between each TC and each of the MTs, and that each CPFs has a storage capacity of 50,000 chassis.
Table 4. Travel times (in seconds) from TCs to CPFs (six-node case).

<table>
<thead>
<tr>
<th>TC Locations</th>
<th>CPF Location and CPF ID</th>
<th>Travel Times (in seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>East Del Amo Rancho Dominguez, CA</td>
<td>Golden Ave, Long Beach, CA 90806 CPF {1}</td>
<td>601</td>
</tr>
<tr>
<td>South Susana Road Rancho Dominguez, CA</td>
<td>Via Oro Ave, Long Beach, CA 90810 CPF {2}</td>
<td>479</td>
</tr>
</tbody>
</table>

Table 5. Travel times (in seconds) from CPFs to TCs (six-node case).

<table>
<thead>
<tr>
<th>CPF Locations</th>
<th>MT Locations</th>
<th>Travel Times (in seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Golden Ave, Long Beach, CA 90806 (CPF {1})</td>
<td>ITS (K-Line) Pier G E, Long Beach, CA 90802, USA</td>
<td>1,044</td>
</tr>
<tr>
<td>Via Oro Ave, Long Beach, CA 90810 (CPF {2})</td>
<td>LBCT (OOCL) Pier F Ave, Long Beach, CA 90802, USA</td>
<td>1,181</td>
</tr>
</tbody>
</table>

Table 6. Travel Times (in seconds) from TCs to MTs (six-node case).

<table>
<thead>
<tr>
<th>TC Locations</th>
<th>MT Locations</th>
<th>Travel Times (in seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>East Del Amo Rancho Dominguez, CA</td>
<td>ITS (K-Line) Pier G E, Long Beach, CA 90802, USA</td>
<td>1,179</td>
</tr>
<tr>
<td>Road Rancho Dominguez, CA</td>
<td>LBCT (OOCL) Pier F Ave, Long Beach, CA 90802, USA</td>
<td>1,057</td>
</tr>
</tbody>
</table>
Figure 11. Centralized Processing of Chassis example. Example schematic for the six-node case. Labels on each edge indicate travel time in seconds between the two connected nodes.

The optimization process is iterated upon while varying the parameter $P$ by steps of thirty seconds. The results are presented in Table 7. In the table one can see that when the chassis retrieval time at MTs and CPFs is identical ($P = 0$), all transactions were routed directly to MTs as there would be no advantage in terms of total time to route the trucks through the CPFs. For the value of $P = 450$ seconds, however, all transactions from TC $\{1\}$ are routed through CPF $\{2\}$. This is expected as the difference in total time between the direct routing from TC $\{1\}$ to the MTs and routing from TC $\{1\}$ to the MTs through CPF $\{2\}$ is 435 seconds (as can be seen in Figure 11). Therefore, as the value of the parameter $P$ increased from 420 to 450 seconds, routing through CPF $\{2\}$ became the preferred option over direct routing to the less efficient MT chassis retrieval location. Similarly for the value of $P = 480$ seconds transactions from TC $\{2\}$ are routed through CPF $\{1\}$. This is also expected as the difference in total time between the direct routing from TC $\{2\}$ to the MTs and routing from TC $\{1\}$ to the MTs through CPF $\{2\}$ is 466 seconds.
These results are also summarized in Figure 12 which shows the number of transactions routed through each CPF as a function of $P$. One can see in the figure that when $P = 450$ seconds, half of the transactions are routed into CPF {2} (rather than being routed directly) and then when $P = 480$ seconds, the other half of the transactions are routed into CPF {1}, and no more transactions are routed directly to the MTs.

### Table 7. Optimal transaction routing (six-node case).

<table>
<thead>
<tr>
<th>Location</th>
<th>Number of Chassis Routed by Location vs. Parameter $P$</th>
<th>Travel Time Increase in seconds due to CPF routing</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC CPF MT</td>
<td>$P = 0$</td>
<td>$P = 450$</td>
</tr>
<tr>
<td>Routing through CPFs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 1 1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1 1 2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1 2 1</td>
<td>0</td>
<td>12,500</td>
</tr>
<tr>
<td>1 2 2</td>
<td>0</td>
<td>12,500</td>
</tr>
<tr>
<td>2 1 1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2 1 2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2 2 1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2 2 2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Direct Routing to MTs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 1</td>
<td>12,500</td>
<td>0</td>
</tr>
<tr>
<td>1 2</td>
<td>12,500</td>
<td>0</td>
</tr>
<tr>
<td>2 1</td>
<td>12,500</td>
<td>12,500</td>
</tr>
<tr>
<td>2 2</td>
<td>12,500</td>
<td>12,500</td>
</tr>
</tbody>
</table>
Figure 12. Optimal transaction routing vs. parameter $P$ (six-node case).
Total number of transactions routed through each CPF or directly to the MT plotted vs. parameter $P$. In this figure: (i) only import transactions are considered; (ii) the model is restricted to six nodes (two TCs, two CPFs and two MTs).

2.3.2.2 Reduced-Node Model Evaluation: Ten Nodes

A second reduced-node case was repeated for a total of ten nodes (three TCs, four CPFs, and three MTs).

The selected locations and travel times between them are shown in Table 8 through Table 10. For this simplified scenario, it is assumed that 5,555 import transactions are required to be completed between each TC and each of the MTs, and that each CPFs has a storage capacity of 50,000 chassis. The optimization process is iterated upon while varying the parameter $P$ in steps of thirty seconds, following the same process as for the six-node case.

The results are presented in Table 11. In the table one can see that when the chassis retrieval time at MTs and CPFs is identical (i.e. $P = 0$) all transactions were routed directly to MTs as there would be no advantage in terms of total time to route the trucks through the CPFs. For the value of $P = 300$ seconds,
however, all transactions from TC \{2\} are routed through CPF \{3\}. This is expected as the difference in total time between the direct routing from TC \{2\} to the MTs and routing from TC \{2\} to the MTs through CPF \{3\} is more than 287 seconds. Therefore, as the value of \( P \) increased from 270 to 300 seconds, routing through CPF \{3\} became the preferred option over direct routing to the less efficient MT chassis retrieval location. Similarly, for the value of \( P = 330 \) seconds, transactions from TC \{1\} are routed through CPF \{6\}. This is also expected as the difference in total time between the direct routing from TC \{1\} to the MTs and routing from TC \{1\} to the MTs through CPF \{6\} is more than 317 seconds. Finally, for \( P = 780 \) seconds, none of the transactions are routed directly to MTs, where they are instead routed through CPFs \{1\}, \{3\}, and \{6\}.

These results are summarized in Figure 13 which shows the number of transactions routed through each CPF as a function of the parameter \( P \). One can see in the figure that when \( P = 300 \) seconds one third of the transactions are routed into CPF \{3\} (rather than being routed directly) and when \( P = 330 \) seconds another third of the transactions are routed into CPF \{6\}. Finally, when \( P = 780 \) seconds, the last third of the transactions are routed through CPF \{1\} and no more transactions are routed directly to the MTs.
Table 8. Travel times (in seconds) from TCs to CPFs (ten-node case).

<table>
<thead>
<tr>
<th>TC Locations</th>
<th>CPF Locations &amp; CPF ID</th>
<th>Golden Ave, Long Beach, CA 90806 CPF {1}</th>
<th>Via Oro Ave, Long Beach, CA 90810 CPF {2}</th>
<th>River Ave, Long Beach, CA 90810 CPF {3}</th>
<th>Long Beach Blvd, Long Beach, CA 90805 CPF {6}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brookhollow Circle Riverside, CA</td>
<td>3,817</td>
<td>3,782</td>
<td>3,855</td>
<td>3,494</td>
<td></td>
</tr>
<tr>
<td>S. Main Street Los Angeles, CA</td>
<td>1,035</td>
<td>1,026</td>
<td>900</td>
<td>890</td>
<td></td>
</tr>
<tr>
<td>W 17th St, Long Beach, CA</td>
<td>493</td>
<td>499</td>
<td>572</td>
<td>446</td>
<td></td>
</tr>
</tbody>
</table>

Table 9. Travel times (in seconds) from CPFs to TCs (ten-node case).

<table>
<thead>
<tr>
<th>CPF Locations</th>
<th>MT Locations</th>
<th>Pacific Container Terminal (COSCO) Harbor Scenic Way, Long Beach, CA 90802</th>
<th>SSA - Pier A Pier C St, Long Beach, CA 90802</th>
<th>SSA (MSC, Zim, SMA/CGM) Pier A Way, Long Beach, CA 90802</th>
</tr>
</thead>
<tbody>
<tr>
<td>Golden Ave, Long Beach, CA 90806, CPF {1}</td>
<td>1,108</td>
<td>748</td>
<td>775</td>
<td></td>
</tr>
<tr>
<td>Via Oro Ave, Long Beach, CA 90810, CPF {2}</td>
<td>1,244</td>
<td>885</td>
<td>886</td>
<td></td>
</tr>
<tr>
<td>River Ave, Long Beach, CA 90810, CPF {3}</td>
<td>1,085</td>
<td>726</td>
<td>774</td>
<td></td>
</tr>
<tr>
<td>Long Beach Blvd, Long Beach, CA 90805 CPF {6}</td>
<td>1,280</td>
<td>921</td>
<td>947</td>
<td></td>
</tr>
</tbody>
</table>
Table 10. Travel times (in seconds) from TCs to MTs (ten-node case).

<table>
<thead>
<tr>
<th>TC Locations</th>
<th>MT Locations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pacific Container Terminal</td>
<td>SSA - Pier A Way, Long Beach, CA 90802</td>
</tr>
<tr>
<td>Harbor Scenic Way, Long</td>
<td>SSA (MSC, Zim, SMA/CGM)</td>
</tr>
<tr>
<td>Beach, CA 90802</td>
<td>Pier A Way, Long Beach, CA 90802</td>
</tr>
<tr>
<td>Pacific Container Terminal</td>
<td>4,458</td>
</tr>
<tr>
<td>Harbor Scenic Way, Long</td>
<td>4,098</td>
</tr>
<tr>
<td>Beach, CA 90802</td>
<td>4,125</td>
</tr>
<tr>
<td>Brookhollow Circle Riverside,</td>
<td>S. Main Street Los Angeles, CA</td>
</tr>
<tr>
<td>CA</td>
<td>1,698</td>
</tr>
<tr>
<td>S. Main Street Los Angeles,</td>
<td>1,338</td>
</tr>
<tr>
<td>CA</td>
<td>1,384</td>
</tr>
<tr>
<td>W 17th St, Long Beach, CA</td>
<td>837</td>
</tr>
<tr>
<td></td>
<td>477</td>
</tr>
<tr>
<td></td>
<td>504</td>
</tr>
</tbody>
</table>
Table 11. Optimal transaction routing (ten-node case).

<table>
<thead>
<tr>
<th>Location</th>
<th>Number of Chassis Routed by Location vs. parameter P</th>
<th>Travel Time Increase in seconds due to CPF routing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TC</td>
<td>CPF</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>Location</td>
<td>Number of Chassis Routed by Location vs. parameter $P$</td>
<td>Travel Time Increase in seconds due to CPF routing</td>
</tr>
<tr>
<td>----------</td>
<td>----------------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>TC</td>
<td>CPF</td>
<td>MT</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Note:

$P = [\text{Average chassis retrieval time at a MT}] - [\text{Average chassis retrieval time at a CPF}]$ (sec)

**Figure 13.** Optimal transaction routing vs. parameter $P$ (ten-node case).

The total number of transactions routed through each CPF or directly to the MT vs. parameter $P$. In this figure: (i) only import transactions are considered; (ii) the model is restricted to ten nodes (three TCs, four CPFs and three MTs).
2.3.2.3 Full model (101-node model)

After verifying that the linear program behaved as expected for the two simplified models used in the reduced-node cases, the full model was analyzed using the same approach. In this case, all of the sixteen potential CPF locations were included, each with the estimated chassis storage capacities provided in Table 3. All seventy-one TCs and fourteen MTs are also included with approximately 50,000 transactions distributed evenly between them (corresponding to approximately fifteen transactions between each TC and each MT). The results are summarized in Figure 14, where it can be seen that when \( P = 0 \) seconds, all of the transactions are routed directly from the TCs to the MTs. However, even with a five-minute increase in efficiency at the CPFs in terms of average chassis retrieval time (corresponding to \( P = 300 \) seconds), approximately half of the transactions are routed through CPFs. The number of transactions that are routed directly from TCs to MTs continues to decrease as the value of the parameter \( P \) increases, up to the point where at \( P = 1,200 \) seconds, virtually no transactions are routed directly to MTs. Table 12 shows the percent utilization of the CPFs for \( P = 1,200 \) seconds. One can see in this case that several of the candidate CPFs are underutilized (e.g. CPF \{5\}, \{10\} and \{11\}). This underutilization indicates that these CPFs are not the best candidates for any of the TCs when there are other choices (i.e. when all of the CPFs are available for use). However, it should be noted these CPFs may still be good candidates when only a limited number of CPF sites are available due to cost or other constraints.
Figure 14. Number of transactions routed through CPFs at optimality, as a function of $P$.

Total number of transactions routed through each CPF or directly to the MT for the optimal solution. In this figure: (i) only import transactions are considered; (ii) all CPFs are available; (iii) the full model is used (including all 101 nodes of TCs, CPFs and MTs).
Table 12. CPF capacity utilization (P=1,200 seconds).

<table>
<thead>
<tr>
<th>CPF ID</th>
<th>Capacity (milliseconds)</th>
<th>Number of transactions routed through CPF</th>
<th>Percent of CPF Capacity Utilized</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7,467</td>
<td>4,182</td>
<td>56%</td>
</tr>
<tr>
<td>2</td>
<td>4,048</td>
<td>1,457</td>
<td>36%</td>
</tr>
<tr>
<td>3</td>
<td>16,779</td>
<td>9,732</td>
<td>58%</td>
</tr>
<tr>
<td>4</td>
<td>6,350</td>
<td>4,382</td>
<td>69%</td>
</tr>
<tr>
<td>5</td>
<td>20,551</td>
<td>206</td>
<td>1%</td>
</tr>
<tr>
<td>6</td>
<td>20,469</td>
<td>10,849</td>
<td>53%</td>
</tr>
<tr>
<td>7</td>
<td>557</td>
<td>557</td>
<td>100%</td>
</tr>
<tr>
<td>8</td>
<td>2,161</td>
<td>2,161</td>
<td>100%</td>
</tr>
<tr>
<td>9</td>
<td>757</td>
<td>757</td>
<td>100%</td>
</tr>
<tr>
<td>10</td>
<td>2,084</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>11</td>
<td>293</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>12</td>
<td>3,192</td>
<td>3,192</td>
<td>100%</td>
</tr>
<tr>
<td>13</td>
<td>1,120</td>
<td>1,120</td>
<td>100%</td>
</tr>
<tr>
<td>14</td>
<td>1,877</td>
<td>1,877</td>
<td>100%</td>
</tr>
<tr>
<td>15</td>
<td>2,258</td>
<td>2,258</td>
<td>100%</td>
</tr>
<tr>
<td>16</td>
<td>6,959</td>
<td>6,959</td>
<td>100%</td>
</tr>
</tbody>
</table>

Note:
(i) Only import transactions are considered
(ii) All CPFs are available
(iii) The full 101-node model is used
2.3.3 Import Only Transactions with Restricted CPF Availability

The next phase in the process is to evaluate the degree of preferential usage for each of the potential CPF candidates. This is an important consideration if the number of potential CPFs is limited due to unavailability. First, it is assumed that only one CPF is available. In order to identify the preferred CPF if only a single CPF location can be used, the problem is solved sixteen times, using only a single different CPF at a time. This is equivalent to the original mathematical formulation with the additional constraints of Equations (19) and (20) applied for the case when $N_{\text{max}} = 1$. The configuration used for TCs and MTs is the same as the one in the original problem, with the exception that a single CPF was included at a time. The results are shown in Figure 15 and Figure 16.

Figure 15 shows the number of transactions routed through each of the candidate CPFs as a function of the parameter P, assuming that only one CPF is available to be used. It can be seen from Figure 15 that the number of transactions routed through a CPF eventually saturates at the capacity of each of the potential CPFs. From this figure, it is seen that the value of the parameter P and the total number of transactions are key drivers in the selection of a single CPF location. For $P \geq 300$ seconds and a total number of transactions greater than 5,000, CPFs {3} and {6} begin to have more transactions routed through them. At this point it becomes more efficient to route some transactions through CPFs {3} and {6}, rather than direct routing to MTs. CPF {5}, on the other hand, does not have the same number of chassis routed through it as CPFs {3} and {6} until P increases by several hundred more seconds.

Figure 16 shows the total travel time for the optimal solution as a function of P, assuming that only one CPF is available to be used. In Figure 16 the optimal solution is shown for each of the CPF routing options. When $P = 500$ seconds the figure shows an observable improvement to the total travel time if trucks are routed through CPFs {3} and {6} as compared to any of the other options. For $P \geq 800$ seconds CPFs {3}, {6} and {5} are clearly superior to all the other CPFs in terms of optimal total travel time.
Figure 15. Optimal transaction routing vs. parameter $P$: one CPF available. The total number of transactions routed through a CPF is plotted against parameter $P$. In this figure: (i) only import transactions are considered; (ii) only one CPF is available at a time; (iii) the full model is used (including all 101 nodes of TCs, CPFs and MTs).
Figure 16. Optimal value of the objective function vs. parameter $P$: one CPF available. The minimum total travel time vs. parameter $P$ at optimality. In this figure: (i) only import transactions are considered; (ii) only one CPF is available at a time; (iii) the full model is used (including all 101 nodes of TCs, CPFs and MTs).

Table 13 shows the complete ranking of all CPFs, if only one CPF can be used. The rankings for Table 13 are obtained for $P = 1,200$ seconds. It is apparent that CPFs {6} and {3} are good candidates if financial resources for only a single CPF are available.
Table 13. Ranking of potential CPF locations if only CFP is available.

<table>
<thead>
<tr>
<th>CPF ID</th>
<th>Address</th>
<th>Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Long Beach Blvd, Long Beach, CA 90805, USA</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>River Ave, Long Beach, CA 90810, USA</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>E Del Amo Blvd, Carson, CA 90746, USA</td>
<td>3</td>
</tr>
<tr>
<td>16</td>
<td>Lomita Blvd, Carson, CA 90745, USA</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>E 213th St, Carson, CA 90746, USA</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>Golden Ave, Long Beach, CA 90806, USA</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>Via Oro Ave, Long Beach, CA 90810, USA</td>
<td>7</td>
</tr>
<tr>
<td>12</td>
<td>Torrance Blvd, Carson, CA 90745, USA</td>
<td>8</td>
</tr>
<tr>
<td>15</td>
<td>S Figueroa St, Wilmington, CA 90744, USA</td>
<td>9</td>
</tr>
<tr>
<td>8</td>
<td>S Sportsman Dr, Compton, CA 90221, USA</td>
<td>10</td>
</tr>
<tr>
<td>14</td>
<td>W Del Amo Blvd, Torrance, CA 90502, USA</td>
<td>11</td>
</tr>
<tr>
<td>10</td>
<td>Alondra Blvd, Paramount, CA 90723, USA</td>
<td>12</td>
</tr>
<tr>
<td>13</td>
<td>W Del Amo Blvd, Torrance, CA 90502, USA</td>
<td>13</td>
</tr>
<tr>
<td>9</td>
<td>Atlantic Ave, Long Beach, CA 90805, USA</td>
<td>14</td>
</tr>
<tr>
<td>7</td>
<td>Long Beach Blvd, Long Beach, CA 90805, USA</td>
<td>15</td>
</tr>
<tr>
<td>11</td>
<td>Alondra Blvd, Paramount, CA 90723, USA</td>
<td>16</td>
</tr>
</tbody>
</table>

2.3.4 Export/Import Integer Linear Formulation with the Full Model

In this section the full 101-node model, including both export and import transactions, is analyzed and the preferred CPF locations are identified. Figure 17 shows the percent improvement in the total travel time when using routing through CPFs as compared to direct routing to the MTs. The optimal solution has been obtained for two values of the parameter $P$: (a) when $P = 600$ seconds, and (b) when $P = 1,200$ seconds, assuming that the total number of export and import transactions are approximately 5,000 and 10,000 respectively.

Figure 17 shows that when there are no CPFs available (i.e. the number of CPFs is zero), then there is no potential for improvement as direct routing through the MTs is the only option. As the number of available CPFs increases from zero to three, the figure shows that there is a significant improvement in using the CPFs as compared to routing directly to the MTs. As expected, the improvement is more dramatic for the higher value of the parameter $P$ ($P = 1,200$ seconds) at 21% as compared to the 7%
improvement for $P = 600$ seconds. Figure 17 also shows that there is no significant improvement to the total travel time if more than three CPFs are utilized, independently of the value of the parameter $P$.

![Figure 17. Number of CPFs used vs. total travel time.](image)

Percent improvement in total travel time at optimality. The improvement is calculated as the percent decrease in total travel time from the case (a) when all transactions are routed to the MTs; to the case (b) when transactions are routed optimally. The percent improvement is plotted against the number of available CPFs. In this figure: (i) both export and import transactions are considered; (ii) two values of the parameter $P$ are studied ($P=600$ seconds, and $P=1,200$ seconds); (iii) the full model is used (including all 101 nodes of TCs, CPFs and MTs).

Table 14 and Table 15 show the complete list of the top five CPF options as a function of the number of CPFs used. In Table 14, $P = 1,200$ seconds. If only one CPF is used, the top three options are CPFs $\{3\}$, $\{6\}$ and $\{16\}$. However, as the number of CPFs used increases from one to two, and then from two to three, the highest ranked options become $\{3,15\}$ and $\{3,6,15\}$ respectively. Similarly, in Table 15, where $P = 600$ seconds, if only one CPF is used, the top three options are CPFs $\{3\}$, $\{15\}$ and $\{6\}$, and as the
number of CPFs used increases the highest ranked options for the cases when two and three CPFs are available become \( \{6, 15\} \) and \( \{3, 6, 15\} \) respectively.

The fact that a difference exists between the individually ranked CPFs and the rankings if combinations of CPFs are used is not surprising. Inclusion of several CPFs allows for the opportunity to select locations spread out in such a way as to offer regional hubs, each of which can be targeted to certain areas. On the other hand, if a single CPF is used, the top rankings then are determined in such a way as to minimize the distance from all of the TCs. In addition, the top ranking CPFs also tend to be located closer to the primary freeways such as the I-710 and the I-110. The optimal CPF locations are shown on the maps of Figure 18 and Figure 19.

As seen in Figure 17, if the number of CPFs used is more than three, there is no significant additional benefit in terms of reduction in total travel time. Hence, for the following analysis a maximum number of three CPFs will be used. To determine which CPFs should be selected to be used in the analysis, the results of Table 14 and Table 15 have been utilized. First, the CPFs were ranked by the number of occurrences within the fifteen potential optimal sets for varying CPF quantities with \( P = 600 \) seconds (i.e. number of times within the set at the top of each of the fifteen lists in Table 15). Figure 20 shows the number of inclusions in the optimal set for each candidate CPF. It is clear that CPFs \( \{3\}, \{6\} \) and \( \{15\} \) are ranked higher than all other options where each is included in fourteen of the fifteen possible optimal sets. The same ranking approach was then used on the results from Table 14 for \( P = 1,200 \) seconds. Table 16 summarizes the rankings of all CPFs for both cases, when \( P = 600 \) seconds and when \( P = 1,200 \) seconds, where it can be seen that the CPFs \( \{3\}, \{6\} \) and \( \{15\} \) were ranked as the top three options for both cases.
Figure 18. Map of the optimal CPF locations ($P = 1,200$ seconds).
(a) One CPF is used (optimal location denoted by the blue pin)
(b) Two CPFs are used (optimal locations denoted by the blue pins)

Figure 19. Map of the optimal CPF locations ($P = 1,200$ seconds).
(a) Three CPFs are used (optimal locations denoted by the blue pins)
(b) Four CPFs are used (optimal locations denoted by the blue pins)
Table 14. Top CPF ranking vs. number of CPFs used: $P = 1,200$ seconds.

<table>
<thead>
<tr>
<th>Number of CPFs Used</th>
<th>CPF IDs for the top 5 options ($P = 1,200$ seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3 6 15</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3 15 6 15</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3 6 15 3 7 15</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3 6 9 15 3 7 15</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3 6 9 12 15</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1 3 6 9 12 15</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1 2 3 6 9 12 15</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1 2 3 4 6 9 12 15</td>
</tr>
<tr>
<td>Number of CPFs Used</td>
<td>CPF IDs for the top 5 options ((P = 1,200) seconds)</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------------------------------------</td>
</tr>
</tbody>
</table>
| 9                  | 1 2 3 4 6 9 12 14 15  
1 2 3 4 6 9 12 13 15  
1 2 3 4 6 9 12 15 16  
1 2 3 4 6 8 9 12 15  
1 2 3 4 6 8 9 14 15 |
| 10                 | 1 2 3 4 6 8 9 12 14 15  
1 2 3 4 6 8 9 12 13 15  
1 2 3 4 5 6 9 12 14 15  
1 2 3 4 5 6 9 12 13 15  
1 2 3 4 6 9 12 14 15 16 |
| 11                 | 1 2 3 4 5 6 8 9 12 14 15  
1 2 3 4 5 6 8 9 12 13 15  
1 2 3 4 6 8 9 12 14 15 16  
1 2 3 4 6 8 9 12 13 15 16  
1 2 3 4 6 7 8 9 12 14 15 |
| 12                 | 1 2 3 4 5 6 8 9 12 14 15 16  
1 2 3 4 5 6 8 9 12 13 15 16  
1 2 3 4 5 6 7 8 9 12 14 15  
1 2 3 4 5 6 7 8 9 12 13 15  
1 2 3 4 6 7 8 9 12 14 15 16 |
| 13                 | 1 2 3 4 5 6 7 8 9 12 14 15 16  
1 2 3 4 5 6 7 8 9 12 13 15 16  
1 2 3 4 5 6 8 9 12 13 14 15 16  
1 2 3 4 5 6 8 9 11 12 14 15 16  
1 2 3 4 5 6 8 9 10 12 14 15 16 |
| 14                 | 1 2 3 4 5 6 7 8 9 12 13 14 15 16  
1 2 3 4 5 6 7 8 9 11 12 14 15 16  
1 2 3 4 5 6 7 8 9 10 12 14 15 16  
1 2 3 4 5 6 7 8 9 11 12 13 15 16  
1 2 3 4 5 6 7 8 9 10 12 13 15 16 |
| 15                 | 1 2 3 4 5 6 7 8 9 10 12 13 14 15 16  
1 2 3 4 5 6 7 8 9 11 12 13 14 15 16  
1 2 3 4 5 6 7 8 9 10 11 12 13 15 16  
1 2 3 4 5 6 8 9 10 11 12 13 14 15 16  
1 2 3 4 5 6 8 9 10 11 12 13 14 15 16 |
Table 15. Top CPF ranking vs. number of CPFs used: \( P = 600 \) seconds.

<table>
<thead>
<tr>
<th>Number of CPFs Used</th>
<th>CPF IDs for the top 5 Options ( (P = 600 \text{ seconds}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3 15 6 12 7</td>
</tr>
<tr>
<td>2</td>
<td>6 15 3 15 7 15 6 16 3 16</td>
</tr>
<tr>
<td>3</td>
<td>3 6 15 3 7 15 3 9 15 4 6 15 3 8 15</td>
</tr>
<tr>
<td>4</td>
<td>3 6 9 15 3 7 9 15 3 4 6 15 3 6 12 15 3 7 12 15</td>
</tr>
<tr>
<td>5</td>
<td>3 4 6 9 15 3 6 9 12 15 3 4 7 9 15 3 7 9 12 15 3 6 9 14 15</td>
</tr>
<tr>
<td>6</td>
<td>2 3 4 6 9 15 2 3 6 9 12 15 3 4 6 9 12 15 3 4 6 9 14 15 3 4 6 9 13 15</td>
</tr>
<tr>
<td>7</td>
<td>2 3 4 6 9 12 15 2 3 4 6 9 14 15 2 3 4 6 9 13 15 2 3 4 7 9 12 15 1 2 3 4 6 9 15</td>
</tr>
<tr>
<td>8</td>
<td>1 2 3 4 6 9 12 15 1 2 3 4 6 9 14 15 1 2 3 4 6 9 13 15 1 2 3 4 7 9 12 15 2 3 4 6 9 12 14 15</td>
</tr>
<tr>
<td>Number of CPFs Used</td>
<td>CPF IDs for the top 5 Options ( P = 600 \text{ seconds} )</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------------------------------------------------</td>
</tr>
</tbody>
</table>
| 9                   | 1 2 3 4 6 9 12 14 15  
1 2 3 4 6 9 12 13 15  
1 2 3 4 6 8 9 12 15  
1 2 3 4 5 6 9 12 15  
1 2 3 4 6 7 9 12 15 |
| 10                  | 1 2 3 4 6 8 9 12 14 15  
1 2 3 4 6 8 9 12 13 15  
1 2 3 4 5 6 9 12 14 15  
1 2 3 4 6 7 9 12 13 15  
1 2 3 4 6 7 9 12 14 15 |
| 11                  | 1 2 3 4 5 6 8 9 12 14 15  
1 2 3 4 5 6 8 9 12 13 15  
1 2 3 4 6 7 8 9 12 14 15  
1 2 3 4 6 7 8 9 12 13 15  
1 2 3 4 5 6 7 9 12 14 15 |
| 12                  | 1 2 3 4 5 6 8 9 12 14 15  
1 2 3 4 5 6 8 9 12 13 15  
1 2 3 4 5 6 8 9 12 14 15 16  
1 2 3 4 5 6 8 9 12 13 14 15  
1 2 3 4 5 6 8 9 11 12 14 15 |
| 13                  | 1 2 3 4 5 6 8 9 12 14 15 16  
1 2 3 4 5 6 7 8 9 12 13 14 15  
1 2 3 4 5 6 7 8 9 11 12 14 15  
1 2 3 4 5 6 7 8 9 10 12 14 15  
1 2 3 4 5 6 7 8 9 12 13 15 16 |
| 14                  | 1 2 3 4 5 6 7 8 9 12 13 14 15 16  
1 2 3 4 5 6 7 8 9 11 12 14 15 16  
1 2 3 4 5 6 7 8 9 10 12 14 15 16  
1 2 3 4 5 6 7 8 9 11 12 13 14 15  
1 2 3 4 5 6 7 8 9 10 12 13 14 15 |
| 15                  | 1 2 3 4 5 6 7 8 9 10 12 13 14 15 16  
1 2 3 4 5 6 7 8 9 11 12 13 14 15 16  
1 2 3 4 5 6 7 8 9 10 11 12 14 15 16  
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15  
1 2 3 4 5 6 7 8 9 10 11 12 13 15 16 |
Figure 20. Ranking of CPF candidate locations ($P = 600$ seconds).
The CPFs are ranked by the number of times each one has been included in the optimal set (based on the results of Table 14 and Table 15). In this figure CPFs \{3\}, \{6\} and \{15\} are the three highest-ranked CPFs.
Table 16. Top CPF ranking export/import.

<table>
<thead>
<tr>
<th>CPF ID</th>
<th>Address</th>
<th>Ranking</th>
<th>$P = 1,200$ seconds</th>
<th>$P = 600$ seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>River Ave, Long Beach, CA 90810, USA</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>S Figueroa St, Wilmington, CA 90744, USA</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>Long Beach Blvd, Long Beach, CA 90805, USA</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>Atlantic Ave, Long Beach, CA 90805, USA</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>E 213th St, Carson, CA 90746, USA</td>
<td>8</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Via Oro Ave, Long Beach, CA 90810, USA</td>
<td>7</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>12</td>
<td>Torrance Blvd, Carson, CA 90745, USA</td>
<td>5</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>Golden Ave, Long Beach, CA 90806, USA</td>
<td>6</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>14</td>
<td>W Del Amo Blvd, Torrance, CA 90502, USA</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>8</td>
<td>S Sportsman Dr, Compton, CA 90221, USA</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>E Del Amo Blvd, Carson, CA 90746, USA</td>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>Long Beach Blvd, Long Beach, CA 90805, USA</td>
<td>13</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>16</td>
<td>Lomita Blvd, Carson, CA 90745, USA</td>
<td>12</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>13</td>
<td>W Del Amo Blvd, Torrance, CA 90502, USA</td>
<td>14</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>10</td>
<td>Alondra Blvd, Paramount, CA 90723, USA</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>11</td>
<td>Alondra Blvd, Paramount, CA 90723, USA</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
</tbody>
</table>

2.3.5 Linear Program Sensitivity Analysis

In this section a sensitivity analysis is performed. The analysis presents the effects on the total travel time of variations in the following quantities:

- The number of CPFs used
- Limitations on CPF capacity
- Total number of transactions
- The ratio of import to export transactions

Table 17 (import transactions only) shows the effect that the number of CPFs used and the CPF capacity limitations have on the total travel time. The results of the table are for the import only situation, when $P = 1,200$ seconds. The table presents several comparisons between cases where the following quantities are varied:
Number of CPFs used (presented in the rows of Table 17). Here comparisons are performed only for the cases when one, two, or three CPFs are used, since using more than three CPFs does not provide significant improvements. All these results are compared to the baseline results of using zero CPFs, which is the case of direct routing to the MTs. The comparisons include the following CPF sets:

- When a single CPF is used, the top five single CPFs are included in the comparison set;
- When two CPFs are used, the top five CPF pairs are included in the comparison set;
- When three CPFs are used, the top five CPF triplets are included in the comparison set;

Capacity of the CPFs (presented in the columns of Table 17). Here comparisons are performed between the cases of unlimited and limited CPF capacity, as estimated previously (Table 3).

The final two columns of Table 17 present the improvements in total travel time when using the CPFs as compared to the baseline scenario (direct routing to the MTs). It is seen that the improvements range from 11% to 21%. When the selected CPFs are restricted to the top ranked only, the improvements are in the range of 16% to 21%. The difference in improvements between the limited CPF capacity and unlimited CPF capacity cases are relatively small, for the top ranked CPFs.
Table 17. CPF rankings vs. CPF capacity: import transactions only.

<table>
<thead>
<tr>
<th>Number of CPFs Used</th>
<th>CPF ID</th>
<th>Value of the objective function</th>
<th>Improvement over Baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Limited CPF Capacity</td>
<td>Unlimited CPF Capacity</td>
<td>Limited CPF Capacity</td>
</tr>
<tr>
<td>0 (Baseline)</td>
<td></td>
<td>3.6731</td>
<td>3.6731</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>3.0765</td>
<td>3.1146</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>3.1522</td>
<td>3.1521</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>3.1992</td>
<td>3.1522</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>3.2011</td>
<td>3.1522</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3.2524</td>
<td>3.1543</td>
</tr>
<tr>
<td>2</td>
<td>3 16</td>
<td>2.9604</td>
<td>2.9251</td>
</tr>
<tr>
<td></td>
<td>3 15</td>
<td>2.9637</td>
<td>2.9311</td>
</tr>
<tr>
<td></td>
<td>6 16</td>
<td>2.9650</td>
<td>2.9311</td>
</tr>
<tr>
<td></td>
<td>1 16</td>
<td>2.9922</td>
<td>2.9536</td>
</tr>
<tr>
<td></td>
<td>3 12</td>
<td>2.9960</td>
<td>2.9604</td>
</tr>
<tr>
<td>3</td>
<td>3 12 15</td>
<td>2.9205</td>
<td>2.8888</td>
</tr>
<tr>
<td></td>
<td>3 6 15</td>
<td>2.9207</td>
<td>2.8889</td>
</tr>
<tr>
<td></td>
<td>3 6 16</td>
<td>2.9225</td>
<td>2.8898</td>
</tr>
<tr>
<td></td>
<td>6 12 15</td>
<td>2.9234</td>
<td>2.9004</td>
</tr>
<tr>
<td></td>
<td>3 14 15</td>
<td>2.9238</td>
<td>2.9104</td>
</tr>
</tbody>
</table>

Similarly, Table 18 (export and import transactions) shows the effect that the number of CPFs used and the CPF capacity limitations have on the total travel time. The results of the table once again correspond to $P = 1,200$ seconds where the following quantities are varied:

- **Number of CPFs used (presented in the rows of Table 18).** Here comparisons are performed only for the cases when one, two, or three CPFs are used. All these results are compared to the baseline results of using zero CPFs, which is the case of direct routing to the MTs. The comparisons include the following CPF sets:
  - When a single CPF is used, the top five single CPFs are included in the comparison set;
  - When two CPFs are used, the top five CPF pairs are included in the comparison set;
  - When three CPFs are used, the top five CPF triplets are included in the comparison set;
• **Capacity of the CPFs (presented in the columns of Table 18).** Here comparisons are performed between the cases of unlimited and limited CPF capacity, as estimated previously (Table 3).

The final two columns of Table 18 present the improvements in total travel time when using the CPFs as compared to the baseline scenario (direct routing to the MTs). It is seen that similar to the import only case, the improvements range from 14% to 21%. When the selected CPFs are restricted to the top ranked only, the improvements are in the range of 16% to 21%. The difference in improvements between the limited CPF capacity and unlimited CPF capacity cases are relatively small, for the top ranked CPFs.

Table 19 shows the CPF ranking when the number of total transactions varies for \( P = 1,200 \). The table shows the results for the case when two CPFs with limited capacity are used. From Table 19 it can be seen that as the total number of transactions grows beyond 30,000, CPF \( \{6\} \) passes CPF \( \{3\} \) as the best possible candidate. This is due to the fact that the capacity of CPF \( \{6\} \) is larger than that of CPF \( \{3\} \), and as the difference between import and export transactions grows beyond 10,000 the benefit of routing transactions to the more optimally located CPF \( \{3\} \) is eventually out-weighed by the higher capacity of CPF \( \{6\} \).
<table>
<thead>
<tr>
<th>Number of CPFs Used</th>
<th>CPF ID</th>
<th>Value of the objective function</th>
<th>Improvement over Baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Total travel time (seconds x $10^7$)</td>
<td></td>
</tr>
<tr>
<td>0 (Baseline)</td>
<td></td>
<td>5.5097</td>
<td>5.5097</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>4.6147</td>
<td>4.6147</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>4.7283</td>
<td>4.6719</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>4.7384</td>
<td>4.7282</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>4.7614</td>
<td>4.7283</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>4.7640</td>
<td>4.7315</td>
</tr>
<tr>
<td>2</td>
<td>3 15</td>
<td>4.3934</td>
<td>4.3877</td>
</tr>
<tr>
<td></td>
<td>6 15</td>
<td>4.4047</td>
<td>4.3966</td>
</tr>
<tr>
<td></td>
<td>3 16</td>
<td>4.4406</td>
<td>4.3966</td>
</tr>
<tr>
<td></td>
<td>1 15</td>
<td>4.4437</td>
<td>4.4303</td>
</tr>
<tr>
<td></td>
<td>6 16</td>
<td>4.4475</td>
<td>4.4406</td>
</tr>
<tr>
<td>3</td>
<td>3 6 15</td>
<td>4.3378</td>
<td>4.3333</td>
</tr>
<tr>
<td></td>
<td>3 7 15</td>
<td>4.3447</td>
<td>4.3333</td>
</tr>
<tr>
<td></td>
<td>3 9 15</td>
<td>4.3458</td>
<td>4.3348</td>
</tr>
<tr>
<td></td>
<td>3 8 15</td>
<td>4.3561</td>
<td>4.3506</td>
</tr>
<tr>
<td></td>
<td>4 6 15</td>
<td>4.3702</td>
<td>4.3655</td>
</tr>
</tbody>
</table>
Table 19. CPF rankings vs. total number of transactions: export and import transactions.

<table>
<thead>
<tr>
<th>Total number of transactions</th>
<th>Number of export transactions</th>
<th>Number of import transactions</th>
<th>Difference between import and export transactions</th>
<th>CPF ID</th>
<th>Value of the objective function Total travel time (seconds x 10^7)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7,500</td>
<td>2,500</td>
<td>5,000</td>
<td>2,500</td>
<td>3 15, 6 15, 1 15, 3 16, 6 16</td>
<td>2.0481, 2.0529, 2.0700, 2.0723, 2.0755</td>
</tr>
<tr>
<td>15,000</td>
<td>5,000</td>
<td>10,000</td>
<td>5,000</td>
<td>3 15, 6 15, 3 16, 6 16, 1 15</td>
<td>4.3934, 4.4047, 4.4406, 4.4437, 4.4475</td>
</tr>
<tr>
<td>22,500</td>
<td>7,500</td>
<td>15,000</td>
<td>7,500</td>
<td>3 15, 6 15, 3 16, 6 16, 3 12</td>
<td>6.4534, 6.4756, 6.5129, 6.5235, 6.5946</td>
</tr>
<tr>
<td>30,000</td>
<td>10,000</td>
<td>20,000</td>
<td>10,000</td>
<td>3 15, 6 15, 3 16, 6 16, 3 12</td>
<td>8.8058, 8.8292, 8.8837, 8.8992, 8.9944</td>
</tr>
<tr>
<td>37,500</td>
<td>12,500</td>
<td>25,000</td>
<td>12,500</td>
<td>6 15, 3 15, 3 16, 6 16, 3 12</td>
<td>10.9243, 10.9705, 10.9824, 10.9842, 11.1492</td>
</tr>
<tr>
<td>45,000</td>
<td>15,000</td>
<td>30,000</td>
<td>15,000</td>
<td>6 15, 6 16, 3 15, 3 16, 3 12</td>
<td>13.3293, 13.3774, 13.4015, 13.4182, 13.6036</td>
</tr>
<tr>
<td>52,500</td>
<td>17,500</td>
<td>35,000</td>
<td>17,500</td>
<td>6 15, 6 16, 3 15, 3 16, 3 6</td>
<td>15.5100, 15.5408, 15.6107, 15.6182, 15.7774</td>
</tr>
<tr>
<td>60,000</td>
<td>20,000</td>
<td>40,000</td>
<td>20,000</td>
<td>6 15, 6 16, 3 15, 3 16, 3 6</td>
<td>17.9485, 17.9864, 18.0569, 18.0770, 18.2081</td>
</tr>
</tbody>
</table>
Figure 21 shows the percent change in total travel time as a function of the ratio of exports and imports for a fixed total number of transactions (approximately 15,000). The results in the figure correspond to the case when two CPFs with limited capacity are used, for $P = 1,200$ seconds. From Figure 21 it can be seen that a 1:1 export/import ratio provides an optimal point, where the allowed reuse of the chassis returned after dropping off exports as chassis for retrieving imports eliminates the impact of the effect of CPF capacity. In addition, one can see that, as would be expected, changing the export/import ratio from 2:1 to 1:1 in this plot shows the same improvement as removing any capacity restrictions on CPFs as noted in Table 18 for the two-CPF case.

![Figure 21. Percent change in total travel time vs. ratio of (export/import) transactions. Percent change in total travel time for optimal solution vs. different values of the ratio of (export transactions)/(import transactions). The zero value in the vertical axis corresponds to the case when this ratio is equal to 1.0.](image)

### 2.4 Summary

In this first study, an analytical framework for modeling and optimization of chassis movements in transportation networks with CPFs was developed which was used to estimate the optimal CPF locations and evaluate the potential for reduction in total travel time to/from ports. The analytical model of the chassis exchange process was then implemented in a case study using the POLB and POLA.
The case study identified sixteen locations in the vicinity of the ports that can be potentially used as CPFs, and evaluated several scenarios of container pick-up/drop-off transactions. The case study used the geographic locations of the sixteen potential CPFs, the locations of the fourteen MTs at the ports which handle container export/import, and the locations of seventy-one TCs in the vicinity of the ports. Results using typical traffic patterns between locations, and typical chassis-related transactions were provided. The models were first validated for reduced-node cases, and then applied to the full-size model which includes all MTs, all potential CPFs and all points of origin for trucks (a total of 101 nodes in the optimization model).

A major advantage of using a CPF for chassis exchange is that the CPF offers a reduced chassis retrieval time, as compared to that at a MT (parameter $P$ defined previously). The results showed that using CPFs provided improvement (reduction) of the total travel time by 7% if the parameter $P = 600$ seconds, and 21% if the parameter $P = 1,200$ seconds. A sensitivity analysis with respect to the number of CPFs employed for chassis exchange, showed that the optimal solution did not improve after more than three CPFs are used, indicating that when real-world financial constraints are imposed, that would likely be an upper threshold of CPF quantities to implement, as building four or more CPFs would not provide any significant return on investment.
3 DYNAMIC DRAYAGE TRUCK SCHEDULING USING CPFS

In the first study of Section 2, the concept of CPFs and the possibility of using them to improve travel time for trucks was evaluated. While the potential benefits at the system/strategic level were estimated in the first study, the question of how best to take advantage of the CPF facilities at the operational level still must be addressed. With further refinement to develop an approach to proactively (and dynamically) schedule drayage operations from a trucking company’s (TC’s) point of view, cost as well as traffic congestion, noise and emissions can be further reduced.

The main objective herein is to develop an analytical framework for dynamic modeling of chassis movements and to investigate optimization techniques for scheduling the tasks and minimizing the total travel time of the drivers from a particular TC’s point of view, when several CPFs are available for use. The methodologies developed can improve TCs’ daily operations, and as a result can improve traffic conditions in the areas surrounding the ports.

In this study, I investigate the temporal components of the CPF concept, where I leverage the results from the first study to determine the number and locations of CPFs, and then optimize the schedules of each truck for a given TC.

Given a particular trucking company (TC), I assume that the set of all daily jobs that need to be completed are known, where each of these jobs consists of moving one container between one of the customer warehouses (WHs) and MTs, or vice-versa, and can be done with or without the use of CPFs. Note that depending upon the destination container configurations and sequence of tasks this would allow for any of the five most common transaction types in MTs presented in Section 1.2.

The formal definition of job for use in this study consists of a required container movement having the following attributes:
i. **Origin.** The origin of a job will be one of the WHs if it transports an export container, or one of the MTs if it is an import container.

ii. **Destination.** Similar to the origin, the destination of a job will be one of the MTs if it is an export container or one of the WHs if it is an import activity.

iii. **Origin Container Configuration.** This refers to the state of the container at the origin (Grounded or Wheeled). It is noted that for my purposes only two states for this attribute are considered.

iv. **Destination Container Configuration.** This refers to the state of the container at the destination (Grounded or Wheeled).

v. **Earliest Allowable Completion Time for the job.**

vi. **Latest Allowable Completion Time for the job.**

The general concept of a series of jobs, i.e. vehicle routing from an individual TC’s point of view including the possible use of CPFs, is illustrated in Figure 22 and Table 20. In Figure 22, a TC needs to complete a set of jobs using the M trucks (or vehicles) available, which are denoted as \( V_1, \ldots, V_M \). It is assumed that the M vehicles available to the TC will be servicing a variety of customer locations and MTs. The L MT locations are given as \( MT_1, \ldots, MT_L \) and the J customer locations which the TC is servicing are generically labeled as warehouses \( WH_1, \ldots, WH_J \).

The set of jobs assigned to \( V_m \) is shown in Table 20, with the resultant path illustrated in Figure 22. In this example, three jobs are to be completed (Job 1 is an export; Jobs 2 and 3 are imports).
• Job 1 consists of picking up a wheeled export container from warehouse $WH_1$ and transporting it to marine terminal $MT_1$, where it will be left in a grounded configuration.

• Job 2 is to pick up a grounded import container from marine terminal $MT_L$ and transport it to warehouse $WH_J$, where it will be left in a wheeled configuration.

• Job 3 is to pick up a grounded import container from marine terminal $MT_1$ and transport it to warehouse $WH_J$, where it will be left in a wheeled configuration. However, the transport truck does not have a chassis, since it left it with the container at warehouse $WH_J$ during completion of Job 2; hence Job 3 will require a visit to the $K^{th}$ CPF, $CPF_K$, to pick up an available chassis, as outlined in Figure 22.

![Figure 22. Example schematic of vehicle routing problem with CPFs.](image)
Table 20. Schedule example for vehicle $V_m$.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Job 1</th>
<th>Job 2</th>
<th>Job 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>(i) Origin</td>
<td>$WH_1$</td>
<td>$MT_L$</td>
<td>$MT_1$</td>
</tr>
<tr>
<td>(ii) Destination</td>
<td>$MT_L$</td>
<td>$WH_1$</td>
<td>$WH_1$</td>
</tr>
<tr>
<td>(iii) Origin Container Configuration</td>
<td>Wheeled</td>
<td>Grounded</td>
<td>Grounded</td>
</tr>
<tr>
<td>(iv) Destination Container Configuration</td>
<td>Grounded</td>
<td>Wheeled</td>
<td>Wheeled</td>
</tr>
<tr>
<td>(v) Earliest Allowable Completion Time for job</td>
<td>8:00 AM</td>
<td>8:00 AM</td>
<td>8:00 AM</td>
</tr>
<tr>
<td>(vi) Latest Allowable Completion Time for job</td>
<td>5:00 PM</td>
<td>5:00 PM</td>
<td>5:00 PM</td>
</tr>
</tbody>
</table>

The general process flow for any vehicle schedule for a given sequence of jobs is shown in Figure 23. For each job there are three basic components: job preparation, job pick-up, and job drop-off.

- **Job preparation** involves assessing whether the vehicle is ready for the current job and, if necessary, picking up a chassis if one is needed for a grounded transaction or dropping off a chassis if the transaction is with a wheeled container.

- **Job pick-up** involves retrieving a wheeled or grounded container from either a WH for an export or a MT for an import.

- **Job drop-off** involves dropping off a wheeled or grounded container from either a MT for an export or a WH for an import.

Job assignment to each of the vehicles and optimization thereof are covered in more detail in the following sections.
Figure 23. General process flow for vehicle schedule.

3.1 Analytical Models and Optimization

In this section, a general analytical framework for the scheduling of jobs for a TC is developed, assuming that CPFs are available to be used. The optimal vehicle scheduling will be identified within this particular framework.

Given:

- the location of the TC, which must complete the particular tasks
- the locations of the MTs, $MT_l, l = 1, ..., L$
- the locations of the customers, or “warehouses” $WH_j, j = 1, ..., J$
- the locations of potential sites for chassis processing facilities $CPF_k, k = 1, ..., K$,
- a set of export and import jobs that need to be completed between $WH_j, j = 1, ..., J$, and $MT_l, l = 1, ..., L$, where each job is determined by its own particular attributes as defined previously in Section 3
- a set of vehicles (trucks) to carry out the jobs
- the maximum allowable work span for any given vehicle
The objective herein is to minimize the weighted combination of:

- the total travel time for all vehicles
- the work span needed to finish all jobs

As defined above, the problem is a multi-objective optimization problem. The purpose of minimizing both total travel time and work span is to provide a more realistic model for the TC’s priorities, where the goal is to minimize (a) the direct hourly costs for completion of jobs, represented by the total travel time for all vehicles, while (b) spreading the jobs as evenly as possible among the vehicle drivers, represented by the work span to finish all jobs. The equal spreading of jobs between drivers is necessary since typically a given staff of drivers is available already to the TC to perform the jobs for the day. Therefore, unequally assigned work would result in staff who, depending upon the pay structure, are either being underutilized (and overpaid) or paid for minimal hours of work so that the TC is not providing a reliable income to their workers and may not be able to maintain trained and available staff.

### 3.1.1 Problem Formulation

The mathematical formulation below is created in order to solve the problem defined in Section 3. The formulation includes models for the variations in travel durations which occur throughout the day for schedule allocation to each of the vehicle drivers. For the purposes of the schedule optimization problem defined herein, the following parameters are assumed to be fixed.

\[
\begin{align*}
J & \quad \text{Number of WHs with which the TC interfaces} \\
K & \quad \text{Number of available CPF locations} \\
L & \quad \text{Number of MTs with which the TC interfaces} \\
N & \quad \text{Number of jobs for TC to perform}
\end{align*}
\]
\( M \) Number of vehicles (trucks) which will work for the given time period

\( TC_1 \) Node representing the TC location

\( WH_j \) Node representing the \( j^{th} \) WH \( j \in \{1, ..., J\} \)

\( CPF_k \) Node representing the \( k^{th} \) chassis processing facility \( k \in \{1, ..., K\} \)

\( MT_l \) Node representing the \( l^{th} \) marine terminal \( l \in \{1, ..., L\} \)

\( WH \) Set of WH nodes

\[ WH \equiv \{WH_1, WH_2, ..., WH_J\} \]

\( CPF \) Set of CPF nodes

\[ CPF \equiv \{CPF_1, CPF_2, ..., CPF_K\} \]

\( MT \) Set of marine terminal nodes

\[ MT \equiv \{MT_1, MT_2, ..., MT_L\} \]

\( V \) Set of vehicles

\[ V = \{V_m\} \quad m = 1, ..., M \]

\( U \) Set of jobs

\[ U = \{u_n\} \quad n = 1, ..., N \]

\( O(u_j) \) Origin of job \( u_j \)

\[ O(u_j) \in MT \cup WH \ \forall j \]

\( D(u_j) \) Destination of job \( u_j \)

\[ D(u_j) \in MT \cup WH \ \forall j \]
\( O_{cfg}(u_j) \) Origin container configuration of job \( u_j \)

\[
O_{cfg}(u_j) \in \{0,1\} \forall j
\]

where

\( O_{cfg}(u_j) = 1 \) represents the case where the container as picked up has a chassis associated with it (i.e. a bobtail must arrive for a “wheeled” pickup), and

\( O_{cfg}(u_j) = 0 \) represents the case where the container as picked up does not have chassis associated with it (i.e. a bobtail with chassis must arrive to for a “grounded” pickup)

\( D_{cfg}(u_j) \) Destination container configuration of job \( u_j \)

\[
D_{cfg}(u_j) \in \{0,1\} \forall j
\]

where

\( D_{cfg}(u_j) = 1 \) represents the case where the container as dropped off has a chassis associated with it (i.e. the bobtail will deliver both chassis and container to complete a “wheeled” drop-off), and

\( D_{cfg}(u_j) = 0 \) represents the case where the container as dropped off does not have chassis associated with it (i.e. the bobtail will deliver only the container and leave with the chassis to complete a “grounded” drop-off)

\( s_{m,i} \) The \( i^{th} \) job in vehicle \( V_m \)’s schedule:

\[
s_{m,i} \in \mathcal{U} \forall m, i \in \mathbb{N}^+\]

\( s_m \) The schedule of vehicle \( V_m \), \( s_m \equiv \{s_{m,1} \ldots s_{m,k}\}\)

\( t_{tot}(s_{m,i}) \) The completion time for job \( s_{m,i} \)

\( T_{max}(u_j) \) Latest allowable completion time for job \( u_j \)

\( T_{min}(u_j) \) Earliest allowable completion time for job \( u_j \)

\( T_{WSmax} \) Maximum allowed work span
\( t_{node}(x_i, x_j, t_k) \) The time to get from node \( x_i \) to node \( x_j \) at time \( t_k \)

\( P(x) \) Processing time for chassis retrieval / drop-off at node \( x, x \in \mathcal{MT} \cup \mathcal{WH} \)

\( T_{wh} \) Time to pick up or drop off wheeled container

\( T_{gnd} \) Time to pick up or drop off grounded container

\( t_0 \) Earliest allowable time for vehicle departure

Note that some assumptions have been made to simplify the modeling process. In this problem, no specific distinction is given as to alternate chassis / container sizes. It is assumed that all of the grounded or wheeled transactions can be accommodated using a single common chassis. This allows for export and import activities to be modeled simply as a directed edge between the appropriate customer and MT (export) or vice versa (import). If a grounded container pick-up is preceded by a grounded container drop-off, the chassis is assumed to be reusable for the next transaction. After dropping off loaded containers at the WHs, it will be necessary to return to that location to pick up the empty and return it to the MT. In this model, this would appear identical to an export transaction. Similarly providing an empty container to a customer for them to load with exports could be modeled as an import transaction in which an empty container is delivered from MT to WH.

The variables \( T(s_m) \) and \( C(s_m) \) introduced in the objective function in Equation (51) below represent the travel time and cost of the schedule of vehicle \( V_m \), and are the building blocks of the multi-objective function described previously. The cost of the schedule \( C(s_m) \) is assumed to be a function of the travel time to complete the schedule. This could include hourly wages as well as other costs to the TC associated with supporting a given vehicle schedule.
The objective function is given by:

\[
\min \sum_{m=1}^{M} C(s_m) + \mu \max_{m=1, \ldots, M} T(s_m)
\]

\[
\text{s.t. } T_{\min}(s_m, i) \leq t_{\text{tot}}(s_m, i) \leq T_{\max}(s_m, i) \quad m = 1, \ldots, M; i = 1, \ldots, |s_m|
\]

\[
|s_m| \geq 1 \quad m = 1, \ldots, M
\]

\[
s_i \cap s_j = \emptyset \quad \forall i \neq j
\]

\[
\bigcup_{m=1}^{M} s_m = \mathcal{U} \quad m = 1, \ldots, M
\]

For a given job, the total completion time is given by

\[
t_{\text{tot}}(s_m, i) = \max \left( t_* + t_{\text{node}}(O(s_m, i), D(s_m, i), t_*) + T_{\text{gnd}} 
\right.

\[
+ (T_{wh} - T_{\text{gnd}}) D_{cf,g}(s_m, i, T_{\min}(s_m, i)) \right) 
\]

\[
t_{\text{tot}}(s_m, 1) = \max \left( t_+ + t_{\text{node}}(O(s_m, 1), D(s_m, 1), t_+) + T_{\text{gnd}} 
\right.

\[
+ (T_{wh} - T_{\text{gnd}}) D_{cf,g}(s_m, 1, T_{\min}(s_m, 1)) \right) 
\]

where

\[
t_* = t_{\text{tot}}(s_m, i-1) + t_{\text{job}}(s_m, i-1, s_m, i, t_{\text{tot}}(s_m, i-1)) + T_{\text{gnd}}
\]

\[
+ (T_{wh} - T_{\text{gnd}}) O_{cf,g}(s_m, i)
\]

\[
t_+ = t_0 + t_{\text{job}}(TC_{\text{job}}, s_m, 1, t_0) + T_{\text{gnd}} + (T_{wh} - T_{\text{gnd}}) O_{cf,g}(s_m, 1)
\]
TC\textsubscript{job} is a “dummy” job defined such that

\begin{align*}
O(TC_{\text{job}}) &= TC_1 \\
D(TC_{\text{job}}) &= TC_1 \\
O_{cfg}(TC_{\text{job}}) &= 1 \\
D_{cfg}(TC_{\text{job}}) &= 1 \\
T_{\text{min}}(TC_{\text{job}}) &= -\infty \\
T_{\text{max}}(TC_{\text{job}}) &= \infty
\end{align*}

\textit{t}_{job}(u_i, u_j, t_k) is the time to get from job \textit{u}_i to job \textit{u}_j at time \textit{t}_k which is given by

\begin{equation}
\textit{t}_{job}(u_i, u_j, t_k) = \begin{cases} 
\textit{t}_{node}(D(u_i), O(u_j), t_k), & (D_{cfg}(u_i) = O_{cfg}(u_i)) \\
\textit{t}_\mp + \textit{t}_{node}(x_{opt}(D(u_i), O(u_j), t_k), O(u_j), t_k + \textit{t}_\mp), & (D_{cfg}(u_i) \neq O_{cfg}(u_i))
\end{cases}
\end{equation}

\textit{x}_{opt}(x_i, x_j, t_k) is the optimal chassis processing location which results in minimal travel / chassis processing time when traveling between nodes \textit{x}_i and \textit{x}_j at time \textit{t}_k

and \ \ \textit{t}_\mp = \textit{t}_{node}(D(u_i), x_{opt}(D(u_i), O(u_j), t_k), t_k) + P \left( x_{opt}(D(u_i), O(u_j), t_k) \right)

Using the recursive formula above, the travel time to complete vehicle \textit{v}_m’s schedule \textit{T}(s_m) is then given by:

\begin{equation}
T(s_m) = t_{tot}(s_m,|s_m|) + t_{job}(s_m,|s_m|, TC_{job}, t_{tot}(s_m,|s_m|)) - t_0 \quad \quad m = 1, \ldots, M
\end{equation}

The cost of the schedule \textit{C}(s_m) is then assumed to be a function of the travel time to complete the schedule as noted below. This function could include the hourly wage of the driver for standard hourly
pay, nonlinear elements to address overtime pay, as well as other costs to the TC associated with
supporting a given vehicle schedule such as average costs due to vehicle maintenance.

\[ C(s_m) = f(T(s_m)) \quad m = 1, \ldots, M \] (63)

### 3.1.2 Optimization methodology

In this project, an optimization methodology is developed to find the optimal solution to the scheduling
problem, as described above. As with most scheduling problems, the problem defined above is NP hard.
In addition, as compared to most typical scheduling problems, there are a few factors which add further
complexity to the current problem, including:

- The current problem is a multi-objective optimization problem. One objective is to minimize the
total cost; the other objective is to minimize the maximum work span of vehicles.
- When the job schedule is such that it includes moving a chassis to/from CPFs, the choice of CPF
is flexible, increasing the size of the potential solution space.
- There is a time window associated with each job.

Although there has been a lot of research in similar routing problems to date, the current exact methods
are limited to problems with only a handful of customers, while real cases can include thousands of
customer and client locations (Labadie, Christian, & Caroline, 2016). Therefore, in order to develop a
scalable approach for this optimization problem, metaheuristic methods are leveraged which can provide
effective and efficient solutions. These problem-independent techniques include approaches which
operate on a single solution such as simulated annealing or tabu searches, as well as approaches which
operate on a set of solutions such as genetic algorithms or particle swarm optimization. Various
metaheuristics were assessed to identify a suitable approach to solve the problem, which have in turn been
adjusted according to the problem at hand to fine-tune its intrinsic parameters. After careful consideration
and evaluation, the genetic algorithm approach was chosen as the metaheuristic to be used.
3.1.3 Genetic Algorithm Overview

With the genetic algorithm, my goal is to minimize the weighted combination of the total travel time for all vehicles and the work span needed to finish all jobs by allocating a fixed set of jobs between WHs and MTs to a given fleet of vehicles. The optimal configuration is described by the job allocation. The problem is such that an ordered set of jobs allocated to each of the vehicles defines any given solution. This ordered set serves as the chromosome in the genetic algorithm. Each of the individual job entries in this set, describing which vehicle is responsible for the job and when it occurs in that vehicle’s schedule, then serve as genes. An example of a set of genes and single chromosome is shown in Figure 24 below.

![Figure 24. Example of chromosomes and genes used in the genetic algorithm.](image)

Given these chromosomes, the process by which the genetic algorithm is implemented to optimize the objective function is shown in Figure 25. The example in Figure 25 is based on scheduling ten vehicles to
complete sixty jobs during a given day. Some of the final settings used in the algorithm including population, crossover percentage, elite count, and termination criteria are indicated in the figure. The basic steps include:

- initialization (where the initial population is generated)
- selection (in which the fitness of the population is evaluated)
- the generation of children based upon the selection criteria
- the implementation of crossover and mutation algorithms

The entire process is then repeated until the termination criteria have been reached.

**Figure 25. Genetic algorithm overview.**
3.1.4 Initial Population

Each of the chromosomes in the initial population for the genetic algorithm was generated using one of four separate algorithms, including two nearest neighbor algorithms and two random permutation algorithms. Note that all four algorithms were built to force the permutations of the jobs spread across the vehicle schedules within a given chromosome to be such that the constraints of Equations (53), (54), and (55) would all be met.

**Nearest Neighbor Algorithm 1**

The first of the chromosomes in the initial population was generated using a nearest neighbor algorithm which equally distributed jobs between all vehicles. This algorithm sequenced through each of the vehicle schedules assigning jobs in the sequence $s_{1,1}, s_{2,1}, \ldots s_{M,1}, s_{1,2}, s_{2,2}, \ldots s_{M,2} \ldots$ until all jobs were assigned, where in each case $s_{m,i}$ was selected such that $t_{job}(s_{m,i-1}, s_{m,i}, t_{tot}(s_{m,i-1}))$ was minimized, and where $s_{m,0} \equiv TC_{job}$ and $t_{tot}(TC_{job}) = t_0$. An example result for this algorithm is shown in Figure 26.
Figure 26. Nearest Neighbor Algorithm 1 example output.
Row $s_m$ represents the schedule of vehicle $V_m$ for the day. Square $u_n$ represents the $n^{th}$ job that the TC has to complete, from a total of $N$ jobs for the day. Job $u_n$ contains all the attributes of a job as defined previously. The $i^{th}$ column represents the $i^{th}$ task in sequence that a vehicle has to perform. The “Nearest Neighbor Algorithm 1” assigns the jobs uniformly to all available vehicles.

Note that in the example above $N/M$ is an integer, which allows equal spreading of jobs between all vehicles. However, the algorithm was written so that if this were not the case the first $N \mod M$ vehicles would be allocated $\lfloor N/M \rfloor$ jobs, while the final $M - (N \mod M)$ vehicles would be allocated $\lfloor N/M \rfloor$. For example if $N = 62$ and $M = 10$, the first $62 \mod 10 = 2$ vehicles are allocated $\lfloor 62/10 \rfloor = 7$ jobs, while the final 8 vehicles are allocated $\lfloor 62/10 \rfloor = 6$ jobs.
Nearest Neighbor Algorithm 2

The second of the chromosomes in the initial population was generated using a nearest neighbor algorithm which assigned one job to each of the vehicles and then assigned all remaining jobs to a single vehicle. This algorithm sequenced through each of the vehicle schedules assigning jobs in the sequence \( s_{1,1}, s_{2,1}, \ldots s_{M,1}, s_{1,2}, s_{1,3}, s_{1,4} \ldots s_{1,N-M+1} \) until all jobs were assigned, such that in each case \( s_{m,i} \) was once again selected to minimize \( t_{job} \left( s_{m,i-1}, s_{m,i}, t_{tot} \left( s_{m,i-1} \right) \right) \). An example result for this algorithm is shown in Figure 27.

![Nearest Neighbor Algorithm 2](image)

**Figure 27. Nearest Neighbor Algorithm 2 example output.**
Row \( s_1 \) represents the schedule of vehicle \( V_1 \) for the day. The “Nearest Neighbor Algorithm 2” assigns only one job to each of the vehicles \( \{V_2, V_3, \ldots V_{10}\} \) and the remaining fifty-one jobs to vehicle \( V_1 \). Note that for simplicity in the figure the entire set of jobs for \( s_1 \) have not been shown.
Random Permutation Algorithm 1

The remaining chromosomes in the initial population were generated using two different algorithms which provide random permutations of the job sequence, with each algorithm generating approximately 50% of the resultant population. In the first random permutation algorithm, jobs were distributed equally across all of the available vehicles. An example result for this algorithm is shown in Figure 28.

Figure 28. Random Permutation Algorithm 1 example output.
Row $s_m$ represents the schedule of vehicle $V_m$ for the day. Square $u_n$ represents the $n^{th}$ job that the TC has to complete, from a total of $N$ jobs for the day. Job $u_n$ contains all the attributes of a job as defined previously. The $i^{th}$ column represents the $i^{th}$ task in sequence that a vehicle has to perform. The “Random Permutation Algorithm 1” assigns the jobs uniformly to all available vehicles.

Note that in the example above $N/M$ is an integer, which allows equal spreading of jobs between all vehicles. However, similar to the case of the first nearest neighbor algorithm, this algorithm was written so that if $N/M$ is not an integer, the first $N$ modulo $M$ vehicles would be allocated $\lceil N/M \rceil$ jobs, while the final $M - (N$ modulo $M)$ vehicles would be allocated $\lfloor N/M \rfloor$. 
Random Permutation Algorithm 2

The other random permutation algorithm first randomly assigned a single job to each of the M vehicles, in order to force meeting the constraint of Equation (53), and then randomly distributed the remaining jobs between all vehicles without any attempt to force an equal distribution. An example result for this algorithm is shown in Figure 29.

Figure 29. Random Permutation Algorithm 2 example output.
Row $s_m$ represents the schedule of vehicle $V_m$ for the day. Square $u_n$ represents the $n^{th}$ job that the TC has to complete, from a total of $N$ jobs for the day. Job $u_n$ contains all the attributes of a job as defined previously. The $i^{th}$ column represents the $i^{th}$ task in sequence that a vehicle has to perform. The “Random Permutation Algorithm 2” first assigns one job to each of the vehicles \{$V_1, V_2, \ldots, V_{10}$\}, and then it assigns the remaining jobs randomly to each vehicle.
3.1.5 **Fitness Function**

In order to compare the quality of different chromosomes within the population, the fitness function for every chromosome represents the weighted sum of the total travel time for all vehicles and the work span needed to finish all jobs, which is calculated according to the algorithm described in Section 3.1.1. In addition, when calculating the fitness, constraint checks according to the optimization algorithm were performed to evaluate each chromosome’s validity. In the case that a chromosome is passed to the fitness function which fails any of the validity checks, the fitness value is not calculated and the chromosome’s fitness value is set to infinity.

3.1.6 **Crossover Function**

The crossover function should be chosen so that when low cost topologies are combined they tend to produce low cost descendants. The crossover function implemented alternates between the two parents’ job sequences at the individual vehicle level to build a solution such that each job is used exactly once, meeting the constraints of Equations (54) and (55). The first vehicle’s schedule for the offspring is copied from that of Parent 1, the second vehicle’s offspring is copied from that of Parent 2, and then this alternating pattern is continued throughout the remainder of the rows. Redundant jobs are then removed and replaced sequentially through the offspring such that each job is used exactly once. An example of a crossover between two parents for \( N = 60 \) and \( M = 10 \) is shown in the figure below. The cases where a redundant job was replaced (such that the child’s schedule for a given vehicle does not exactly match one of the parents) are shown in red.
3.1.7 Mutation Function

Three different mutation functions were used with equal probability each time the mutation function was called. The first mutation function involved moving a job, whereas the second two mutation functions involved swapping of jobs rather than moving them. Note that for each of these three mutation functions the result could be moving / swapping jobs within a given vehicle’s schedule, or a moving / swapping of jobs between two different vehicles’ schedules.

**Mutation Function 1**

In the first of the mutation functions, a single job was selected at random and moved into a random location. The job to be moved was selected by first randomly selecting one of the vehicle schedules with more than one job, and then randomly selecting among the jobs for that specific schedule. The destination was selected in a similar fashion by first selecting a vehicle schedule at random (this time allowing for any of the vehicle schedules to be selected regardless of jobs currently in the schedule), and then randomly selecting the location in the schedule into which the job would be inserted. An example of this algorithm is shown in Figure 31, where the job which is moved between parent and child is highlighted in red.
Mutation Function 1 example.

In the first of the swapping mutation functions, two jobs were randomly selected and swapped with all jobs having equal likelihood of selection. An example of this algorithm is shown in Figure 32, where the jobs which are swapped between parent and child are highlighted in red.

Mutation Function 2

In the first of the swapping mutation functions, two jobs were randomly selected and swapped with all jobs having equal likelihood of selection. An example of this algorithm is shown in Figure 32, where the jobs which are swapped between parent and child are highlighted in red.
**Mutation Function 3**

In the final mutation algorithm, a swap was performed between the two jobs \((s_{m,i} \text{ and } s_{n,j})\) with the largest total job-to-job travel times according to Equation (64) below.

\[
\max_{m=1,\ldots,M, n=1,\ldots,M} t_{job}(s_{m,i-1}, s_{m,i}, t_{tot}(s_{m,i-1})) + t_{job}(s_{m,i}, s_{m,i+1}, t_{tot}(s_{m,i})) + t_{job}(s_{n,j-1}, s_{n,j}, t_{tot}(s_{n,j-1})) + t_{job}(s_{n,j}, s_{n,j+1}, t_{tot}(s_{n,j}))
\]

An example of this algorithm is shown in Figure 33, where the jobs which are swapped between parent and child are highlighted in red. Note that the key difference between this mutation and the previous result from Mutation Function 2 is that \(s_{8,2}\) and \(s_{10,2}\) in Figure 32 were selected at random whereas \(s_{5,5}\) and \(s_{6,3}\) selected by Mutation Function 3 had the two largest values for job-to-job travel times in the parent chromosome as defined in Equation (64) above.

![Figure 33. Mutation Function 3 example.](image)
3.1.8 Termination Function

The termination function used for the algorithm included a maximum number of total iterations which would be performed as well as a maximum number of iterations which would be performed in a row if the improvement in the cost was less than or equal to a fixed tolerance level.

3.2 Case Study Model Implementation

The analytical models and optimization methods described in Section 3.1 are applied in the case study to evaluate the impact of CPF usage while the total cost and maximum work span for the TC are minimized in a multi-objective cost function.

Real life simulation scenarios are developed using past, current and projected data from the LB/LA port area. The simulation scenarios are used to evaluate and compare two scenarios: base operations, and chassis and container movements with CPFs in the loop.

- **Base Operations:** The base operations replicate the current practices in the LB/LA port area. Here, the simulation experiments provide baseline data for total travel times and predicted work spans of vehicles for the modeled TC in the existing situation.

- **CPF for Chassis Operations:** Simulation scenarios are developed and executed based on the results of the optimization procedure described above. The results of the various simulation scenarios with CPFs in the loop are compared to the base operations (when CPFs are not being used), and the improvements of using CPFs are quantified.

The case study used the MTs in the POLB and POLA complex, one representative TC, a number of WHs, and leveraged the same potential locations for CPFs in the vicinity of the ports as in the first study. The representative TC was selected at random from the set of TCs used in the first study, while a representative set of WHs covering the area of interest for the TC was generated using a list similar to the list of addresses in Section 2.2.2. In addition, the results of the first study showed that most of the
improvement in the optimal solution was achieved when three CPFs were used. Therefore, only the three top CPF locations that were identified in the previous project were used in the current case study (CPF \{3\}, \{6\}, and \{15\}) as shown in Figure 19(a).

### 3.2.1 Jobs

Jobs were selected randomly using the WH and MT locations noted previously by using the following assumptions:

- Export to Import ratio of 1:2
- Total number of jobs for the selected TC in one day is set to 60
- Wheeled vs. non-wheeled containers randomly selected with 50% probability of either for both WH and MT locations
- Minimum and maximum completion times for all jobs set at 6:00 am and 12:00 am, allowing for a range of 18 hours within which the jobs could be completed

The jobs selected for the case study using this approach are shown in Figure 34, which represents a map of the area covered with WH nodes identified as cyan boxes, MTs identified as black circles, the TC identified as a green circle, the imports shown as blue dotted lines, and the exports shown as blue solid lines, where the export and import jobs have been plotted on separate maps in order to increase legibility.
Figure 34. Map of jobs used in case study.
(a) Maps of Export Jobs where the symbols represent the TC, WH and MT locations
(b) Maps of Import Jobs where the symbols represent the TC, WH and MT locations

3.2.2 Travel Time Between Locations

The models for travel times between the TC and all of the WHs, CPFs and MTs were developed using the GDM API. In order to create a complete model of the area of interest to support the case study, time-varying models covering a twenty-four hour period for a typical workday needed to be generated between all locations. For this particular case study this includes \((1 + J + K + L)^2 = (1 + 70 + 3 + 14)^2 = 7,744\) possible routes. In addition, the profiles of travel times from the GDM API are classified into two categories, using (a) pessimistic and (b) optimistic assumptions. Therefore, to generate a time-varying model for all possible routes including optimistic and pessimistic times would require \(7,744 \times 2 \times 24/\delta t\)
individual queries, where $\delta t$ is the time step of interest. Even for a time step of thirty minutes this will result in 743,424 queries to the GDM API, where the cost associated with its usage, at $0.01 per query, makes querying all routes for an entire day cost prohibitive.

The main purpose for generating a time-varying model for all possible routes is to make an assessment of the effectiveness of a complex approach, which considers the variation of traffic conditions throughout the day. With this goal in mind, and given that making 743,424 or more queries to the GDM API is practically impossible, it was necessary to construct a suitable model to represent the daily variations of travel times between any two arbitrary locations, using a limited set of queries.

### 3.2.2.1 Single Sample per Route Least Squares Model

In order to generate the simplified model, a set of sixteen representative trips was considered initially, defined by the origin/destination pairs as shown in Table 21. These sixteen characteristic trips were then queried for both optimistic and pessimistic times at a thirty-minute spacing over a twenty-four hour period from midnight on Wednesday, July 24 to midnight Thursday, July 25, 2019, resulting in $16 \times 2 \times 48 = 1,536$ queries. The locations selected are listed in Table 21 and then shown on Figure 35. In Table 21, locations are color coded by the location type for the TC, WHs and MTs per the legend. In Figure 35, the origins are shown in green, and the destinations in red, with all other nodes shown as x’s.
Table 21. Origin destination pairs for daily variation estimates.

<table>
<thead>
<tr>
<th>Trip</th>
<th>Origin</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TC {1} 2321 East Del Amo Rancho Dominguez, CA</td>
<td>WH {30} 1483 W Via Plata St Long Beach, CA</td>
</tr>
<tr>
<td>2</td>
<td>TC {1} 2321 East Del Amo Rancho Dominguez, CA</td>
<td>WH {44} 8800 Slauson Ave, Pico Rivera, CA</td>
</tr>
<tr>
<td>3</td>
<td>TC {1} 2321 East Del Amo Rancho Dominguez, CA</td>
<td>WH {48} 851 E Watson Center Rd, Carson, CA</td>
</tr>
<tr>
<td>4</td>
<td>TC {1} 2321 East Del Amo Rancho Dominguez, CA</td>
<td>MT {14} 701 New Dock Street Terminal Island, CA</td>
</tr>
<tr>
<td>5</td>
<td>WH {3} 2059 Belgrave Ave Huntington Park, CA</td>
<td>WH {30} 1483 W Via Plata St Long Beach, CA</td>
</tr>
<tr>
<td>6</td>
<td>WH {3} 2059 Belgrave Ave Huntington Park, CA</td>
<td>WH {44} 8800 Slauson Ave, Pico Rivera, CA</td>
</tr>
<tr>
<td>7</td>
<td>WH {3} 2059 Belgrave Ave Huntington Park, CA</td>
<td>WH {48} 851 E Watson Center Rd, Carson, CA</td>
</tr>
<tr>
<td>8</td>
<td>WH {3} 2059 Belgrave Ave Huntington Park, CA</td>
<td>MT {14} 701 New Dock Street Terminal Island, CA</td>
</tr>
<tr>
<td>9</td>
<td>WH {8} 131 E Gardena Blvd Gardena, CA</td>
<td>WH {30} 1483 W Via Plata St Long Beach, CA</td>
</tr>
<tr>
<td>10</td>
<td>WH {8} 131 E Gardena Blvd Gardena, CA</td>
<td>WH {44} 8800 Slauson Ave, Pico Rivera, CA</td>
</tr>
<tr>
<td>11</td>
<td>WH {8} 131 E Gardena Blvd Gardena, CA</td>
<td>WH {48} 851 E Watson Center Rd, Carson, CA</td>
</tr>
<tr>
<td>12</td>
<td>WH {8} 131 E Gardena Blvd Gardena, CA</td>
<td>MT {14} 701 New Dock Street Terminal Island, CA</td>
</tr>
<tr>
<td>13</td>
<td>MT {1} 1048 Pier G E, Long Beach, CA</td>
<td>WH {30} 1483 W Via Plata St Long Beach, CA</td>
</tr>
<tr>
<td>14</td>
<td>MT {1} 1048 Pier G E, Long Beach, CA</td>
<td>WH {44} 8800 Slauson Ave, Pico Rivera, CA</td>
</tr>
<tr>
<td>15</td>
<td>MT {1} 1048 Pier G E, Long Beach, CA</td>
<td>WH {48} 851 E Watson Center Rd, Carson, CA</td>
</tr>
<tr>
<td>16</td>
<td>MT {1} 1048 Pier G E, Long Beach, CA</td>
<td>MT {14} 701 New Dock Street Terminal Island, CA</td>
</tr>
</tbody>
</table>

**LEGEND**

- **TC {1}** Location of the Trucking Company used in this study
- **WH {X}** Location of Warehouse X (of 70)
- **MT {Y}** Location of Marine Terminal Y (of 14)
Figure 35. Map of jobs used in daily traffic variation model. The symbols in the graph of Figure 35 represent the origin and destination points as explained in the legend, based on the locations given in Table 21.

For the sixteen trips noted above, optimistic and pessimistic travel times were calculated at a thirty-minute spacing.

One example of the driving paths between an origin/destination pair is shown in Figure 36. The example origin/destination pair corresponds to the fourth row (Trip 4) of Table 21, between trucking company TC {1} (origin) and marine terminal MT {14} (destination).
In this query the Google Directions API © was used rather than the GDM API in order to provide the complete set of directions between two points rather than only the optimistic and pessimistic travel durations. The Google Directions API © was queried to provide directions between TC \{1\} and MT \{14\} during peak traffic, for which it suggested three alternate paths and a range of travel time estimates for each. For example, it is seen that the optimistic travel time for the path highlighted in blue is twenty minutes, whereas the pessimistic travel time estimate for the blue path is forty minutes. Note that in general the Google Directions and Distance Matrix APIs © only provide estimates for typical passenger car routing, and there may be times when trucks cannot follow the same routes that are available to typical passenger car traffic. In addition, one can see that, at this peak travel time, the optimal route suggested by the Google Directions API © (blue path) actually shows a slightly longer pessimistic travel time estimate than one of the other alternative paths (forty minutes vs. thirty-five minutes). This is worth noting only for the fact that the data as delivered for pessimistic and optimistic travel times from the GDM API may have some inherent noise due to the underlying algorithms and routing approaches. Using the thirty-minute spacing, Figure 37 shows the optimistic and pessimistic daily profiles for traveling between TC \{1\} and MT \{14\}, and all sixteen origin and destination pairs from Table 21 are shown in the summary figures which follow.
Figure 36. Peak predicted travel time (recommended route is highlighted in blue).
Figure 37. Daily travel variation optimistic / pessimistic estimates (Trip 4).

Figure 38 and Figure 39 show a summary of the optimistic and pessimistic travel time estimates for all sixteen trips. The black trace in Figure 38 provides an average optimistic travel time estimate over all sixteen trips. Similarly, the black trace in Figure 39 provides an average pessimistic travel time estimate over all sixteen trips. The traces in Figure 40 have been constructed to show the average travel time (i.e. the mid-point between the optimistic and the pessimistic estimates) for each of the sixteen trips. The black trace in Figure 40 provides an average of the average travel time estimate over all sixteen trips. In general, it can be seen that there is a similar pattern across most of the trips in which the travel time profiles have peaks during rush hour periods, typically from 5:00 to 7:00 a.m., and from 2:00 to 6:00 p.m. It is noted that the peaks are much more pronounced in the pessimistic models than they are in the optimistic models.
Figure 38. Optimistic travel time estimates.

Figure 39. Pessimistic travel time estimates.
The next phase in generating the simplified model is the construction of a simple function $g$ which can be used by the genetic algorithm, whenever a coarse prediction of the time-varying traffic conditions is needed. For this purpose, first a total of 7,744 queries were made to GDM API, to obtain the travel times at midnight for all 7,744 possible routes between origins and destinations used in the case study, as described at the beginning of Section 3.2.2. Next the representative sixteen routes for which data was queried for the entire day were used and an average of the average travel time profiles was obtained, which is plotted as the solid black trace in Figure 40, and is denoted by $\bar{y}$ in the equations below. The function $g$ was then constructed as shown below. Function $g$ is used to provide coarse estimates of the travel time $t_{node}(x_m, x_p, t_k)$ between two arbitrary nodes $x_m$ and $x_p$ at time $t_k$. Given a pair of nodes $x_m, x_p$ the function $g$ will use the average travel time at midnight between $x_m$ and $x_p$ as explained above, denoted by $t_{node}(x_m, x_p, 0)$. Using the raw $t_{node}(x_m, x_p, 0)$ value provided by the GDM API, and the complete set of data for travel time estimates throughout the day that were generated by the sixteen
representative trips, a least squares model for the function \( g(x_m, x_p, t_k) \) was created. The function, \( g \), was modeled to be of the form

\[
g(x_m, x_p, \delta t * j) = \alpha + \beta t_{node}(x_m, x_p, 0) + (\gamma + \rho t_{node}(x_m, x_p, 0))\bar{y}_j \quad j = 0, ..., n \tag{65}
\]

where

\[
n = 24 * 3600 / \delta t
\]

\[
\bar{y}_j = \frac{\sum_{k=1}^{B} y_{k,j}}{B} \quad j = 0, ..., n \tag{66}
\]

\[
y_{k,j} \equiv \text{The typical duration from the GDM API for trip } k \text{ at time sample } j \quad k = 1, ..., B \quad j = 0, ..., n
\]

\( B \) is the total number of representative trips (16 in this case)

and \( \alpha, \beta, \gamma, \text{ and } \rho \) are model parameters to be determined.

The step-size \( \delta t \) of the measurements is represented in seconds. For the data above all values were recorded at a thirty-minute (1,800 second) spacing giving \( n = 24 * \frac{3,600}{1,800} = 48 \).

In order to find the best estimates for model parameters \( \alpha, \beta, \gamma, \text{ and } \rho \) a least squares fit was performed, solving Equation (67) below for the best estimate, in a least squares sense, of \( x \) given by \( \hat{x} = (A^T A)^{-1} A^T y \). The fitted values were then used in Equation (65) and applied to each of the sixteen representative trip values at midnight to provide a comparison between the route specific detailed estimates and those based upon the least squares fit and the average data in Equation (65).

The comparisons between the simplified model and the individual data sets are plotted in Figure 41 through Figure 48.

\[
y = Ax \tag{67}
\]

where
\[ x = \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \rho \end{bmatrix} \] (68)

\[ y = \begin{bmatrix} y_1 \\ \vdots \\ y_B \end{bmatrix} \] (69)

\[ y_k = \begin{bmatrix} y_{k,0} \\ \vdots \\ y_{k,n} \end{bmatrix} \] (70)

\[ \bar{y} = \begin{bmatrix} \bar{y}_0 \\ \vdots \\ \bar{y}_n \end{bmatrix} \] (71)

\[ A = \begin{bmatrix} 1_n & y_{1,0}1_n & \bar{y} & y_{1,0}\bar{y} \\ 1_n & y_{2,0}1_n & \bar{y} & y_{2,0}\bar{y} \\ \vdots & \vdots & \vdots & \vdots \\ 1_n & y_{B-1,0}1_n & \bar{y} & y_{B-1,0}\bar{y} \\ 1_n & y_{B,0}1_n & \bar{y} & y_{B,0}\bar{y} \end{bmatrix} \] (72)
Figure 41. Daily travel variation vs. least squares fit (1-2).
Figure 42. Daily travel variation vs. least squares fit (3-4).
Figure 43. Daily travel variation vs. least squares fit (5-6).
Figure 44. Daily travel variation vs. least squares fit (7-8).
Figure 45. Daily travel variation vs. least squares fit (9-10).
Figure 46. Daily travel variation vs. least squares fit (11-12).
Figure 47. Daily travel variation vs. least squares fit (13-14).
Figure 48. Daily travel variation vs. least squares fit (15-16).
In order to assess the quality of the least squares fit, Figure 49 shows the percentage residual error for the sixteen trips as the difference between the red and the blue traces of the sixteen plots shown in Figure 41 through Figure 48. In general, this model provided a fairly representative depiction of the general daily variations; however it resulted in large residual error increases during rush hour periods, typically from 5:00 to 7:00 a.m., and from 2:00 to 6:00 p.m.

![Figure 49. Residual of daily travel variation vs. least squares fit based model.](image)

### 3.2.2.2 Multiple Sample per Route Least Squares Model

In order to provide a more accurate model of the daily variations, an approach leveraging additional sample points, beyond a single sample point at midnight, for each of the potential network origin destination pairs was developed. Although the entire set of $7,744 \times 2 \times 24/\delta t$ samples is too large to query for the entire network, some smaller set of fixed samples, $q$, for each of the locations could be sampled, such that the total $7,744 \times 2 \times q$ samples could be leveraged in a segmented least squares model. In this section a model was developed to perform a segmented least squares fit using $q$ sample points, and
the optimal points for a given value of \( q \) are identified. In the segmented least squares approach, the representative routes were once again used to generate \( \bar{y} \), (an average of the average travel time profiles as demonstrated in Figure 40), however in this case a least squares fit is generated for each segment between sample points. Using this, the function \( h \) was constructed as shown below. Function \( h \), similar to function \( g \) in Equation (65), provides estimates of the travel time \( \hat{t}_{\text{node}}(x_m, x_p, t_k) \) between two arbitrary nodes \( x_m \) and \( x_p \) at time \( t_k \). Given a pair of nodes \( x_m, x_p \) the function \( h \) will include inputs from the GDM API at \( q \) discrete points \( d_1 \) through \( d_q \) where the GDM API average travel times are denoted by \( t_{\text{node}}(x_m, x_p, d_1) \) through \( t_{\text{node}}(x_m, x_p, d_q) \). An example of these segments for \( q = 4 \) samples taken at times \( d_1 \) to \( d_4 \) is illustrated in the figure below:

![Segmented least squares fit example](image)

**Figure 50. Segmented least squares fit example.**

Using these samples and the complete set of data for travel time estimates from a set of representative trips, a least squares model for the function \( h(x_m, x_p, d_1, ..., q, t_k) \) was created. The function, \( h \), was modeled to be of the form
\[ h(x_m, x_p, d_1, ..., d_q, j_\delta t) \quad j = 0, ..., n \]

\[
\begin{cases}
\alpha_1 + \beta_{1,1} t_{node}(x_m, x_p, d_1) + \beta_{1,2} t_{node}(x_m, x_p, d_2) + \cdots \quad d_1 \leq j_\delta t < d_2 \\
(\gamma_1 + \rho_{1,1} t_{node}(x_m, x_p, d_1) + \rho_{1,2} t_{node}(x_m, x_p, d_2)) \bar{y}_j, \\
\alpha_2 + \beta_{2,1} t_{node}(x_m, x_p, d_2) + \beta_{2,2} t_{node}(x_m, x_p, d_3) + \cdots \quad d_2 \leq j_\delta t < d_3 \\
(\gamma_2 + \rho_{2,1} t_{node}(x_m, x_p, d_2) + \rho_{2,2} t_{node}(x_m, x_p, d_3)) \bar{y}_j, \\
\vdots \\
\alpha_q + \beta_{q,1} t_{node}(x_m, x_p, d_q) + \beta_{q,2} t_{node}(x_m, x_p, d_{q+1}) + \cdots \quad d_q \leq j_\delta t < d_{q+1} \\
(\gamma_q + \rho_{q,1} t_{node}(x_m, x_p, d_q) + \rho_{q,2} t_{node}(x_m, x_p, d_{q+1})) \bar{y}_j,
\end{cases}
\]  

(73)

where

\[ n = 24 \times 3600 / \delta t \]

\[ \bar{y}_j = \frac{\sum_{k=1}^{B} y_{k,j}}{B} \quad j = 0, ..., n \]

\[ y_{k,j} \equiv \text{The typical duration from the GDM API for trip } k \text{ at time } j \]

\[ k = 1, ..., B \]

\[ j = 0, ..., n \]

\[ B \text{ is the total number of trips included in the generation of the average data set} \]

\[ d_1 \equiv 0 \]

\[ d_{q+1} \equiv 24 \times 3600 \]

\[ t_{node}(x_m, x_n, d_{q+1}) \equiv t_{node}(x_m, x_n, d_1) \]

and \( \alpha, \beta, \gamma, \text{ and } \rho \) are model parameters to be determined

Note that \( d_1 \equiv 0 \) as the values at midnight were already collected. In addition, the value at time \( d_{q+1} \) was not collected, as it was assumed that the average travel time at midnight would be identical on two consecutive days. The least squares fit was then performed, solving Equation (74) below as was done for (67) previously.
\[ y = Ax \] (74)

where

\[
x = \begin{bmatrix}
\alpha_1 \\
\vdots \\
\alpha_q \\
\beta_{1,1} \\
\vdots \\
\beta_{q,1} \\
\beta_{1,2} \\
\vdots \\
\beta_{q,2}
\end{bmatrix}
\] (75)

\[
y = \begin{bmatrix}
y_1 \\
\vdots \\
y_B
\end{bmatrix}
\] (76)

\[
y_k = \begin{bmatrix}
y_{k,0} \\
\vdots \\
y_{k,n}
\end{bmatrix}
\] (77)

\[
l_n^z = l_n(z) = \begin{cases}
1, d_z \leq \delta t + 0 < d_{z+1} \\
0, otherwise
\end{cases}
\] (78)

\[
\bar{y} = \begin{bmatrix}
\bar{y}_0 \\
\vdots \\
\bar{y}_n
\end{bmatrix}
\] (79)

\[
A = [A_1 \ A_2]
\] (80)
\[ A_1 = \begin{bmatrix} l_n^1 & l_n^q & y_{1,d_1} l_n^1 & y_{1,d_q} l_n q & y_{1,d_2} l_n^1 & y_{1,d_q} l_n^1 & y_{1,d_{q+1}} l_n^1 \\ l_n^1 & l_n^q & y_{2,d_1} l_n^1 & y_{2,d_q} l_n q & y_{2,d_2} l_n^1 & y_{2,d_q} l_n^1 & y_{2,d_{q+1}} l_n^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ l_n^1 & l_n^q & y_{B-1,d_1} l_n^1 & y_{B-1,d_q} l_n q & y_{B-1,d_2} l_n^1 & y_{B-1,d_q} l_n^1 & y_{B-1,d_{q+1}} l_n^1 \\ l_n^1 & l_n^q & y_{B_1,d_1} l_n^1 & y_{B_1,d_q} l_n q & y_{B_1,d_2} l_n^1 & y_{B_1,d_q} l_n^1 & y_{B_1,d_{q+1}} l_n^1 \end{bmatrix} \]  

\[ A_2 = \begin{bmatrix} l_n^1 \circ \overline{y} & l_n^q \circ \overline{y} & y_{1,d_1} l_n^1 \circ \overline{y} & y_{1,d_q} l_n q \circ \overline{y} & y_{1,d_2} l_n^1 \circ \overline{y} & y_{1,d_q} l_n^1 \circ \overline{y} & y_{1,d_{q+1}} l_n^1 \circ \overline{y} \\ l_n^1 \circ \overline{y} & l_n^q \circ \overline{y} & y_{2,d_1} l_n^1 \circ \overline{y} & y_{2,d_q} l_n q \circ \overline{y} & y_{2,d_2} l_n^1 \circ \overline{y} & y_{2,d_q} l_n^1 \circ \overline{y} & y_{2,d_{q+1}} l_n^1 \circ \overline{y} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ l_n^1 \circ \overline{y} & l_n^q \circ \overline{y} & y_{B-1,d_1} l_n^1 \circ \overline{y} & y_{B-1,d_q} l_n q \circ \overline{y} & y_{B-1,d_2} l_n^1 \circ \overline{y} & y_{B-1,d_q} l_n^1 \circ \overline{y} & y_{B-1,d_{q+1}} l_n^1 \circ \overline{y} \\ l_n^1 \circ \overline{y} & l_n^q \circ \overline{y} & y_{B_1,d_1} l_n^1 \circ \overline{y} & y_{B_1,d_q} l_n q \circ \overline{y} & y_{B_1,d_2} l_n^1 \circ \overline{y} & y_{B_1,d_q} l_n^1 \circ \overline{y} & y_{B_1,d_{q+1}} l_n^1 \circ \overline{y} \end{bmatrix} \]  

\[ y_{k,d_{q+1}} \equiv y_{k,d_1} \equiv y_{k,0} \]

and \( \circ \) denotes the Hadamard or entrywise product.

In Section 3.2.2.1 sixteen example routes were used to both generate the least squares fit coefficients and evaluate the accuracy of the fits. In this case separate training and evaluation data were used to evaluate fits, and the quantity of representative routes was increased from sixteen to fifty. The original sixteen routes were used, but an additional thirty-four were selected by randomly choosing origin and destinations from the set of potential nodes in the case study resulting in a total of fifty routes. These fifty routes were then queried at a thirty-minute spacing over an entire day from midnight Wednesday, July 24, 2019 to midnight Thursday, July 25, 2019. Then twenty-five were chosen for training, while twenty-five were chosen for evaluation. Once the source data was generated, evaluations were performed to determine which indices gave the best possible fit for \( q = 1 \) through \( q = 6 \), where the upper limit of six samples was set as a budgetary constraint based upon the costs of the GDM API queries noted previously. The optimal samples for which to perform the fit at each value of \( q \) were determined by doing the following:
• Recursively cycling through all possible permutations of $q$ samples from the forty-eight possible thirty-minute samples collected for the representative runs

• For each permutation performing the least squares fit using the training routes, applying the resultant fit coefficients and sample locations from the given permutation to generate fit estimates on the evaluation data, and calculating the root-mean-square error (RMSE) in the given estimate using the GDM API values from the evaluation data set as truth data

• Sorting the resultant error of all possible permutations for the given value of $q$ and selecting the sample locations with the lowest RMSE for the evaluation data.

The set of optimal sample times (i.e. minimum RMSE in the evaluation data set) from for $q = 1$ through $q = 6$ are as shown in Table 22. The reduction in residual error as a function of sample quantity is shown in Figure 51.

Table 22. Optimal sample times vs. sample quantity.

<table>
<thead>
<tr>
<th>$q$</th>
<th>Sample Times</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12:00 AM</td>
</tr>
<tr>
<td>2</td>
<td>12:00 AM</td>
</tr>
<tr>
<td>3</td>
<td>12:00 AM</td>
</tr>
<tr>
<td>4</td>
<td>12:00 AM</td>
</tr>
<tr>
<td>5</td>
<td>12:00 AM</td>
</tr>
<tr>
<td>6</td>
<td>12:00 AM</td>
</tr>
</tbody>
</table>
Each of the individual plots are shown in Figure 52 through Figure 64 and a summary showing the residual error as a percentage of the travel route for all 25 evaluation runs is included in Figure 65.
Figure 52. Daily travel variation vs. segmented least squares fit for $q = 6$ (1-2).
Figure 53. Daily travel variation vs. segmented least squares fit for $q = 6$ (3-4).
Figure 54. Daily travel variation vs. segmented least squares fit for q = 6 (5-6).
Figure 55. Daily travel variation vs. segmented least squares fit for $q = 6$ (7-8).
Figure 56. Daily travel variation vs. segmented least squares fit for $q = 6$ (9-10).
Figure 57. Daily travel variation vs. segmented least squares fit for q = 6 (11-12).
Figure 58. Daily travel variation vs. segmented least squares fit for $q = 6$ (13-14).
Figure 59. Daily travel variation vs. segmented least squares fit for $q = 6$ (15-16).
Figure 60. Daily travel variation vs. segmented least squares fit for $q = 6$ (17-18).
Figure 61. Daily travel variation vs. segmented least squares fit for $q = 6$ (19-20).
Figure 62. Daily travel variation vs. segmented least squares fit for q = 6 (21-22).
**Figure 63.** Daily travel variation vs. segmented least squares fit for q = 6 (23-24).
Figure 64. Daily travel variation vs. segmented least squares fit for $q = 6$ (25).

Figure 65. Evaluation data set residual error for $q = 6$. 
In general, this model provided a much better fit with respect to the training data than that included in Figure 49 of Section 3.2.2.1. The single largest outlier, occurring just before five hours at approximately 30%, is actually due to a small phase offset in the fit for trip 23 (see Figure 63). In order to use this model for the complete data set in the case study, fit parameters per Equation (75) were generated using the optimal fit locations $d_1, ..., d_6$ and all 50 of the representative runs which had been sampled for an entire day. Then the complete node set (consisting of 7,744 potential routes) was queried using the GDM API at these six optimal times. For any route between nodes $x_m$ and $x_p$, these six optimal times were then used as inputs into the function $h(x_m, x_p, d_1, ..., d_6, t_k)$.

### 3.2.2.3 Adding Noise to the model

In order to evaluate methods for dynamically rescheduling a TC’s drayage operations it was necessary to differentiate between: average daily travel times which could be used for planning and job assignment prior to the day, more accurate real-time travel duration estimates which could be provided during the day (i.e. estimated travel times provided by mapping software which take into account current traffic conditions), and true travel times. To do this a representative stochastic model for time-varying errors in the predicted travel durations was developed. An estimate of the distribution of the travel duration error as a function of time was created using the same fifty representative routes sampled at a thirty-minute period as were used in Section 3.2.2.2. The optimistic and pessimistic travel times over the full day as a percentage of the average travel duration at each sample throughout the day for these runs is plotted in Figure 66, which includes each of the individual runs as well as the average bounds across all fifty runs.
Figure 66. Optimistic and pessimistic bounds as percentage of average travel duration.

The average bounds were then used as a basis for the distributions of the random errors in the complete network travel durations throughout the day, where truth data with representative travel duration error distributions could be generated by adding random errors to the average daily travel times which were used for planning and job assignment prior to the day. Then a smaller proportional error could be added onto the truth data to generate real-time travel duration estimates which were more accurate than the original daily estimates in order to assess the potential benefits of real-time dynamic re-allocation of jobs between vehicles. In order to inject random errors with the approximate error bounds proportional to those above, I generated a function which would take the average route estimate as an input and then add normally distributed data with a standard deviation at each sample point equal to half of the width of the distance between the optimistic and pessimistic bounds from Figure 66 which could be scaled by an adjustable scale factor. For example, if the average duration at the sample time of interest was 1,000 seconds and a scale factor of two was used, the standard deviation in the error at midnight was 200 seconds (10% of the average time multiplied by a scale factor of two), 500 seconds at 8 am (25% of the
average time multiplied by a scale factor of two), etc. following a scaled version of the average curves shown in Figure 66. However, although the data generated using this approach would have the desired error magnitude for each individual time step, it would not be representative of true changes in travel duration over time which would not be expected to have any high frequency content. Therefore, I generated a low-pass filter which could be applied to the time series data. To generate realistic predicted travel duration variations I used a low-pass equiripple FIR with a passband equivalent to a four-hour period, a ripple of 1 dB and stopband attenuation of 60 dB using the Parks-McClellan algorithm. The magnitude response of the filter is shown in Figure 67. This filter was selected in order to avoid any phase shift in the input data (to avoid moving the key rush hour periods) while making sure that variations which would be generated in the data would not have unrealistically high frequency content.

![Figure 67. Low-pass filter for noise elimination.](image)

After the addition of the band-limited noise, negative travel times were removed from the network if they existed, and then the entire network was modified to reinforce the triangle equality (see Section 3.2.2.4). This processing, which was necessary in order to ensure realistic data for each time step across the network, was nonlinear and required some tuning of the data afterwards to enforce the desired scaling of
the output noise, and to avoid introduction of any biases. Figure 68 and Figure 69 show the effects of the additional processing, where the results prior to tuning diverge from the ideal. To generate the results below, the noisy data sets for the entire network over a twenty-four hour period were generated for noise scale factors from 0.1 up to five. Then for each set of data, the RMSE of the fit between the original and noisy data sets was plotted as a percentage of the desired input standard deviation (Figure 68), and the mean of the noisy data was plotted as a percentage of the original data’s mean (Figure 69). In both cases, the results diverge from the ideal where the noise levels were less than 30% of the desired for all of the noise scale factors, and the mean increased by more than 50% as the desired noise scale factor increased to five. Therefore, in order to adjust for the nonlinearity in the noise injection and triangle fixing processes, I had to adjust the noise magnitude and data set bias as a function of the desired noise scale factor. A gradient descent algorithm was employed to determine the mean and standard deviation tuning values as a function of the desired noise scale factor with a 5% tolerance. Figure 68 and Figure 69 show the results of implementing the additional scaling, where after tuning, the injected noise and resultant mean were both within 5% of the desired values for any noise scaling values between zero and five times the reference standard deviation envelope shown in Figure 66.
Figure 68. Noise injected into travel time estimates versus desired.

Figure 69. Average travel time estimate versus original.
In Figure 70, this same tuning was used and the RMSE of the fit between the average data and noisy data across the entire network as a function of time is plotted for four different noise scale factors. The data is plotted as a percentage of the average value to allow for a more direct comparison with the baseline noise shown in Figure 66, where it can be seen that the result for a noise scale factor of one in Figure 70 matches the baseline noise envelope in Figure 66, and each of the other three scale factors adjust the RMSE up or down as desired.

![Figure 70. Route duration error magnitudes vs. time as percent of average value. Route duration error magnitudes are RMSE resulting from the comparison of noisy data to average data for all nodes used in case study at each point in time. In this figure: (i) noise scale factor is scaling of baseline noise envelope as described in Section 3.2.2.3](image)

**3.2.2.4 Triangle Fixing**

As noted in Section 3.2.2.3, the addition of noise into the travel durations throughout the network model requires that the data be reprocessed to enforce the triangle equality for each time step. To do this required finding a solution to the metric nearness problem which, in general, deals with optimally
restoring metric properties to distance measurements that happen to be nonmetric due to measurement errors. This was done by implementing a triangle fixing algorithm based upon an iterative projection method which finds a nearest set of distances to the source data that satisfy the properties of a metric. Although there is no analytic solution to the metric nearness problem, the problem does lend itself to a convex formulation, which makes developing algorithms for it much easier. In this case the triangle fixing was performed for ℓ₂ metric nearness according to the algorithm provided by (Brickell, Dhillon, Sra, & Tropp, 2008). As an input to the triangle fixing algorithm, it was necessary to construct a complete cost matrix for the data which requires triangle fixing at time \( t_k \), \( C^*_{t_k} \) below, which includes the travel times between all nodes in the network at time \( t_k \).

\[
C^*_{t_k} = \begin{bmatrix}
C^*_{TC_1,TC_1,t_k} & C^*_{TC_1,WH,t_k} & C^*_{TC_1,CPF,t_k} & C^*_{TC_1,MT,t_k} \\
C^*_{WH,TC_1,t_k} & C^*_{WH,WH,t_k} & C^*_{WH,CPF,t_k} & C^*_{WH,MT,t_k} \\
C^*_{CPF,TC_1,t_k} & C^*_{CPF,WH,t_k} & C^*_{CPF,CPF,t_k} & C^*_{CPF,MT,t_k} \\
C^*_{MT,TC_1,t_k} & C^*_{MT,WH,t_k} & C^*_{MT,CPF,t_k} & C^*_{MT,MT,t_k}
\end{bmatrix} (83)
\]

where

\[
C^*_{TC_1,TC_1,t_k} = t_{node}(TC_1, TC_1, t_k) = 0
\]

\[
C^*_{TC_1,WH,t_k} = \begin{bmatrix} t_{node}(TC_1, WH_1, t_k) & \ldots & t_{node}(TC_1, WH_J, t_k) \\
\vdots & \ddots & \vdots \\
\end{bmatrix} (84)
\]

\[
C^*_{MT,MT,t_k} = \begin{bmatrix} t_{node}(MT_1, MT_1, t_k) & \ldots & t_{node}(MT_1, MT_L, t_k) \\
\vdots & \ddots & \vdots \\
t_{node}(MT_L, MT_1, t_k) & \ldots & t_{node}(MT_L, MT_L, t_k) \\
\end{bmatrix}
\]

Then for time interval \( t_k \), the \( nxn \) matrix \( C^*_{t_k} \) (where \( n = 1 + J + K + L \)) was input to the triangle fixing algorithm below along with the tolerance value in seconds, \( \kappa \), to generate the triangle equality compliant matrix \( C_{t_k} \), where for all results provided herein, the tolerance \( \kappa \) was set at 0.1 seconds.
\( triangle \_fixing \_\ell_2(C^*_t, \kappa) \)

**Input:** Travel Duration Matrix \( C^*_t \), tolerance \( \kappa \)

**Output:** \( C_t = \arg\min_{X \in \mathcal{M}_N} \|X - C^*_t\|_2 \) (where \( \mathcal{M}_N \) is the set of all \( nxn \) distance matrices)

\[
\begin{align*}
    d_{ij} &= C^*_t(i, j) \quad \text{for} \ 1 \leq i \leq n, 1 \leq j \leq n \\
    e_{ab} &\leftarrow 0 \text{ for } 1 \leq a < b \leq n \\
    (z_{abc}, z_{bca}, z_{cab}) &\leftarrow 0 \text{ for } 1 \leq a < c < b \leq n \\
    \delta &\leftarrow 1 + \kappa \\
    \text{while } (\delta < \kappa) \\
    &\quad \text{foreach triangle inequality } (a, b, c) \\
    &\quad \quad v \leftarrow d_{ac} + d_{cb} - d_{ab} \\
    &\quad \quad \theta^* \leftarrow \frac{1}{3}(e_{ab} - e_{ac} - e_{cb} - v) \\
    &\quad \quad \theta \leftarrow \max\{\theta^*, -z_{acb}\} \\
    &\quad \quad e_{ab} \leftarrow e_{ab} - \theta, e_{ac} \leftarrow e_{ac} + \theta, e_{cb} \leftarrow e_{cb} + \theta, \\
    &\quad \quad z_{acb} \leftarrow z_{acb} + \theta \\
    &\quad \text{end foreach} \\
    &\quad \delta \leftarrow \text{sum of changes in the } e_{ab} \text{ values} \\
    &\quad \text{end while} \\
    &\text{return } C_t = C^*_t + E \text{ (where } E = \begin{bmatrix} e_{11} & \cdots & e_{1n} \\
                                      \vdots & \ddots & \vdots \\
                                      e_{n1} & \cdots & e_{nn} \end{bmatrix} \)
3.2.2.5 Adding Accidents

In order to generate representative accidents in the data set, the nodes included within the case study were first grouped in order to determine which sets of nodes would likely use the same highways to connect to the others. A k-means clustering algorithm was used on the cost matrices output from the triangle fixing algorithm in Section 3.2.2.4 for cluster quantities \( k = 2, \ldots, 20 \). The cluster size for grouping the data was then selected such that it minimized the Davies-Bouldin index, \( C_k \), as shown in Equations (85) through (87), which evaluates the within-cluster scatter divided by the distance from the nearest cluster, averaged across all clusters. (David & Bouldin, 1979)

\[
\min_k \{ C_k \} \tag{85}
\]

\[
C_k = \frac{1}{k} \sum_{i=1}^{k} \max_{j \neq i} \{ D_{i,j} \} \tag{86}
\]

where

\[
D_{i,j} = \frac{\bar{d}_i + \bar{d}_j}{d_{i,j}} \tag{87}
\]

\( \bar{d}_i \) is the average distance between the centroid and each point in the \( i^{th} \) cluster

\( \bar{d}_j \) is the average distance between the centroid and each point in the \( j^{th} \) cluster

\( d_{i,j} \) is the Euclidean distance between the \( i^{th} \) and \( j^{th} \) cluster centroids

The result for \( k = 2, \ldots, 20 \) is shown in Figure 71, where the optimal number of clusters was found to be 13.
The 13 clusters for the network can be seen in Figure 72 below.
For each of these clusters, the node nearest to the cluster centroid was selected as a representative sample for that cluster, and the Google Directions API © was queried to provide turn by turn directions between each of the representative samples including $13 \times 12 = 156$ inter-cluster routes. These directions were then reviewed to determine usage of key freeways in both directions including the following:

Figure 72. Network clusters for $k = 13$. 
- I-710 N
- I-710 S
- CA-91 W
- CA-91 E
- I-110 N
- I-110 S
- I-105 W
- I-105 E
- CA-47 N
- CA-47 S
- I-405 N
- I-405 S
- I-605 N
- I-605 S

The assumption was then made that the representative cluster to cluster routes would apply for any cluster to cluster interactions, such that, for example, if there was an accident on the I-710N, any of the representative inter cluster routes which included the I-710N would have the same change in bias of the route time, as would any other origin and destination pairs which belonged to the same origin and destination clusters as the representative route which included using the I-710 N.

In order to determine the most likely accident times, the frequency of accidents as a function of time of day was found using the traffic collision database from Los Angeles from 2010 to the present (LAPD, 2018). This data was used to generate an accident probability as a function of time which is shown in Figure 73. This could then be used to randomly inject accidents throughout the day in a representative fashion.
The accidents were then injected as a step change in travel duration followed by a linear decrease back to the original traffic model over a predefined interval.

Examples of the combination of the noise, filtering, and randomized accidents (including the application of the triangle fixing algorithm across the entire network) can be seen in Figure 74 through Figure 77. In each case they include the following:

- Initial day estimates: the initial day estimates are used for daily route planning and are based upon the average data for the GDM API using the algorithms in Section 3.2.2.2.
- Truth data: the truth data was created using the initial day estimates plus noise as described in Section 3.2.2.3 plus a random accident as described above.
- Real-time estimates: the real-time estimates are more accurate than the initial day estimates and are generated by adding a small amount of noise to the truth data.
In the cases below the accident was injected as a two-hour delay which cleared after a period of two hours. Note that this two-hour delay is similar to many of the representative threshold cases used in previous work developed in order to model secondary accident rates as studied by (Moore II, Giuliano, & Cho, 2004) and (Sun & Chilukuri, 2010).

![Figure 74. Example of noise and I-710 N accident.](image-url)
Figure 75. Example of noise and CA-91 W accident.

Figure 76. Example of noise and I-110 N accident.
Figure 77. Example of noise and CA-91 E accident.
3.2.2.6 Modeling Change in Prediction Accuracies

Lastly a function was generated which would blend between two different accuracies over a given time window. This is useful for modeling the near term accuracy of a solution (which is still not perfect), and its gradual reduction in accuracy into the future. In the examples in Figure 78 through Figure 80 the real-time estimates were merged over a four-hour window starting at 10 am, such that at 2 pm (as viewed from 10 am) they had the same accuracies as the initial day estimates.

![Graph](image)

**Figure 78.** Decrease in real-time estimate accuracy over time with I-110N accident.
3.2.3 Additional Time Settings

In the genetic algorithm the initial time for the vehicle departures ($t_0$) was set at 6:00 am. For the maximum allowable driver work span, $T_{WSmax}$, the initial runs of the model used twelve hours in order to allow for a larger solution space when running the genetic algorithm. However, for the scenarios defined
herein the actual work span on the optimized solutions was typically much less than this upper limit with typical values less than or equal to eight hours.

In order to account for the additional difficulty of picking up and dropping off grounded containers as compared to wheeled containers, the time to pick up or drop off wheeled containers \( T_{wh} \) was set at five minutes whereas the time to pick up or drop off grounded containers \( T_{gnd} \) was set at fifteen minutes.

Lastly, as noted previously in Section 2.2.5 it is assumed that there will be an inherent advantage in using the CPFs over the MTs for chassis retrieval and drop-off. The difference between the processing time at MTs and that at CPFs is referred to as “Additional Processing Time” and denoted by \( P \). Based on similar considerations as originally noted in in Section 2.2.5, the processing time for chassis retrieval / drop-off, \( P(y) \), was set at five minutes for \( y \in CPF \), while the processing time \( P(x) \), was set at either fifteen or twenty-five minutes for \( x \in MT \) depending upon the specific scenario being evaluated. The two values selected for MT processing times result in \( P = 600 \) and \( P = 1,200 \) seconds of “Additional Processing Time” respectively.

### 3.2.4 Optimization and Genetic Algorithm Specific Settings

For all case studies and sensitivity analyses in Section 3.3.1 through Section 3.3.3, the cost of each schedule, \( C(s_m) \) of Equation (63), was assumed to be equal to the travel time to complete vehicle \( v_m \)'s schedule \( T(s_m) \). This results in a simplified objective function, Equation (51), of the form:

\[
\min \sum_{m=1}^{M} T(s_m) + \mu \max_{m=1,\ldots,M} T(s_m)
\]  

(88)

In this form, the objective function minimizes the weighted sum of the total travel time needed to complete all vehicles’ schedules, \( \sum_{m=1}^{M} T(s_m) \), and the maximum work span across all of the vehicles’
schedules, \( \max_{m=1, \ldots, M} T(s_m) \). As \( \mu \) increases, the objective function will force more uniformity across the vehicles’ schedules, so that all vehicles have approximately the same work span. A sensitivity analysis was performed as a function of \( \mu \) in Section 3.3.3. In that analysis, a reasonable value which maintained uniformity across vehicles’ schedules for the given job set was identified as \( \mu = 1 \), which was in turn used as the nominal setting in the other case studies.

In addition, the following configuration was used for the genetic algorithm.

- The population was set at sixty
- 500 generations were run unless a stall limit of no improvement for 200 generations in a row was reached
- ten of each subsequent generation were generated by directly passing on the best “elite” solutions from the previous generation without any modification
- thirteen of each subsequent generation were created using the crossover function
- thirty-seven of each subsequent generation were created using the mutation function

The genetic algorithm parameters, including population, number of total generations and stall generations as well as the composition of each generation by direct passing of fittest parents, crossover, and mutation were arrived at through early experimentation when developing the genetic algorithm. During this experimentation values in the following range were evaluated:
- Number of Generations: 500-2,000
- Stall Limit: 200-600
- Population Size: 60-240
- Elite Count: 1-25
- Crossover fraction: ¼ - ¾

In order to assess their impact, the genetic algorithm parameters noted above were varied while repeating the genetic algorithm optimization for the same case. For each genetic algorithm run, the case included sixty jobs, assumed \( P = 1,200 \) seconds, and used the average travel durations for the entire network generated as described in Section 3.2.2.2. The number of generations, stall limit, and population size all showed approximately linear increases in computational time as the values increased. For example when varying the population size from 500 to 2,000 generations, the computational time ranged from 7.5 to 31 minutes when running with an Intel® Core™ i7-6820HQ CPU @ 2.70GHz processor and 16.0 GB RAM using Matlab version R2016a. However, this significant change in computational time resulted in small improvements in the overall result. As shown in Figure 81 the generation and stall limit increase to 2,000 and 600 respectively resulted in less than a 0.5% improvement, while Figure 82 shows no obvious improvement when running with an increased population. Increasing the elite count showed a small reduction in overall computational time; however, as shown in Figure 83, an elite count of ten gave the optimal solution within the range selected, and increasing the elite count too high comes with the risk of allowing the model to become trapped at a local optimum. Lastly, as shown in Figure 84, crossover fractions of \( \frac{1}{4} \) and \( \frac{1}{2} \) both gave similar results; however, a crossover fraction of \( \frac{1}{4} \) was selected, in general, as it gave the best performance over a larger number of generations. For the given case study parameters, the final configuration noted in the previous paragraph was found to provide good solutions in several minutes where no significant improvements were seen with increased generations or populations sizes.
Figure 81. Genetic algorithm total generations impact evaluation.

Figure 82. Genetic algorithm population size impact evaluation.
Figure 83. Genetic algorithm population elite count impact evaluation.

Figure 84. Genetic algorithm crossover fraction impact evaluation.
3.3 Case Study Simulation Results

This section presents the results of several simulation scenarios for the case study, based on the optimization formulation and case study model implementation described previously.

3.3.1 Genetic Algorithm Evaluation

A typical genetic algorithm output is shown in Figure 85 and Figure 86. Figure 85 shows one of the ten vehicle schedules, namely that of vehicle $V_2$, and Figure 86 shows the complete chromosome with all vehicle schedules. These outputs were generated for $M = 10$ vehicles for a particular day, when they have to complete $N = 60$ jobs. In the figures:

- each solid blue line represents a specific job
- each dotted line indicates an additional move to CPF or MT in order to perform a chassis exchange or a relocation between jobs
- numbers next to each node indicate the order of arrival between nodes
- a green circle indicates the TC
- red circles indicate CPFs
- cyan squares indicate the WHs
- black circles indicate the MTs
Figure 85. Example of truck schedule ($s_2$) used in the case study. The schedule was generated by the genetic algorithm for vehicle $V_2$, with $M = 10$ and $N = 60$. 
During the initial genetic algorithm assessment, various settings were evaluated before converging on the settings defined in Sections 3.1.4 and Section 3.2.4. One of these early improvements to the algorithm was the addition of the nearest neighbor solution to the initial population. In the original version of the
genetic algorithm the initial population was created by using only random sequencing and allocations of the jobs to the vehicles. Early in the development the initial population was updated to include the nearest neighbor solutions as defined in Section 3.1.4. The improvement in the conversion of the algorithm can be seen in Figure 87. In the figure it can be seen that in the original algorithm the solution from the first generation was more than 10% of the best solution, whereas in the updated algorithm with the nearest neighbor added to the initial population the initial solution is only off by 2% from the best solution calculated after 500 generations. In the case of the updated algorithm, the initial generation’s solution is almost as good as the final solution from the previous algorithm. In addition, this analysis shows what benefit the optimization program is providing over a simple heuristic nearest neighbor approach, where there is approximately a 2% improvement. That indicates that a pretty good solution can be found, using a nearest neighbor heuristic, relatively quickly in the case that the computing resources are limited such that the larger optimization cannot be performed.
3.3.2 Impact of CPFs

In order to assess the impact of the use of CPFs, several cases were run. These cases were run for $P = 600$ seconds and $P = 1,200$ seconds, over a range of a number of jobs from $N = 15$ to $N = 60$.

- **Cases with CPFs.** When CPFs are present and can be utilized (designated as cases “w/ CPFs” on the plots), chassis exchanges were allowed to occur in both CPFs and MTs. The average processing time for chassis at a CPF was set at 300 seconds. The average processing time at a MT was set at either 900 seconds (when the parameter $P = 600$ seconds), or at 1,500 seconds (when the parameter $P = 1,200$ seconds).
• **Cases without CPFs.** When CPFs are not available (designated as cases “w/o CPFs” on the plots), all necessary chassis exchanges are forced to occur at one of the MTs. The same average processing time as before of either 900 seconds or 1,500 seconds has been used.

In addition, these cases were run for two scenarios to allow for better comparison to the previous study. The main factors in the two scenarios are the attributes of successive jobs, in particular attribute (iv) of \( \text{Job}_k \), and attribute (iii) of \( \text{Job}_{k+1} \), as defined in Section 3.

• **Scenario 1.** In this scenario, if attribute (iv) of \( \text{Job}_k \) is the same as attribute (iii) of \( \text{Job}_{k+1} \), i.e. if the container configuration at the destination point of \( \text{Job}_k \) is identical to the container configuration at the origin point of \( \text{Job}_{k+1} \), then the truck driver does not need to travel to a CPF or to a MT to pick up or drop off a container.

• As an example, for the jobs defined in Table 21 and represented in Figure 22, it can be seen that:
  - Attribute (iv) of \( \text{Job}_1 \) shows that the container at the destination is left in a grounded configuration; hence the truck departs the destination point of \( \text{Job}_1 \) carrying a chassis, and arrives at the origin point of of \( \text{Job}_2 \) with a chassis.
  - Attribute (iii) of \( \text{Job}_2 \) shows that the container at the origin point of of \( \text{Job}_2 \) is grounded; hence the truck will load the container on the chassis. The truck does not need to perform a drop-off or pick-up chassis operation while transitioning from \( \text{Job}_1 \) to \( \text{Job}_2 \).
  - Attribute (iv) of \( \text{Job}_2 \) shows that the container will be left at the destination point of \( \text{Job}_2 \) in a wheeled configuration; hence the truck leaves the destination point of \( \text{Job}_2 \) as a bobtail.
  - Attribute (iii) of \( \text{Job}_3 \) shows that the container at the origin point of of \( \text{Job}_3 \) is grounded, hence the truck after completing \( \text{Job}_2 \) will need to pick up a chassis before starting \( \text{Job}_3 \). In this particular example, the truck will pick up a chassis at CPF\( K \) as shown in Figure 22.
• **Scenario 2.** In this scenario a chassis operation (pick-up or drop-off) is forced to occur between jobs, regardless of container configuration. The reason for this scenario is to provide a better comparison to the previous study, where every transaction was grounded and required the truck driver to stop at a CPF or MT in order to complete a job. Consequently, Scenario 1 above, results in a smaller number of chassis transactions than what was modeled and reported in the first study. Scenario 2 will provide a limiting upper bound for what might occur in the real world when there are additional limitations of chassis usage, and there are job-to-job differences in chassis configuration needs (i.e. not all jobs are forty-foot containers of the same type).

The results for Scenario 1 are shown in Figure 88 through Figure 90. The results for Scenario 2 are shown in Figure 91 through Figure 93.

Scenario 1 allows for direct routing between origin and destination (i.e. a CPF is used only if the container configuration at the origin does not match the container configuration at the destination). In Figure 88 the greatest improvement for the cost function due to the use of CPFs is seen to occur for small ratios of jobs to vehicles. When the number of jobs is small, the improvement is approximately 5% (for both $P = 600$ and $P = 1,200$ seconds). For large ratios of jobs to vehicles there is only a small benefit at approximately 1% for sixty jobs (allocated between ten vehicles). Figure 89 and Figure 90 show a qualitative representation of the relative difference between the solutions with small and large ratios of jobs to vehicles. Note that in the figures any time a CPF is being used, it is because it is saving time as compared to the use of a MT for the chassis exchange. Therefore, the greater number of paths connected to CPFs is a qualitative indication of the potential for additional reductions in total travel time as compared to the case where CPFs are not available.

Scenario 2 forces a chassis operation to occur between jobs, regardless of container configuration. The chassis operation may occur at a CPF or at a MT. Figure 91 shows that under Scenario 2 for values of $P$ between 600 and 1,200 seconds, there is an improvement of 18% to 30% when the number of jobs is
small, and an improvement of 15% to 23% when the number of jobs is large. It is noted that this analysis is from the point of view of a single TC (i.e. at the operational level), but the observed behavior is similar to what was observed during the analysis at the strategic level in Section 2 which showed 7% - 21% improvements in total travel time for the overall network.

Note that there is not a direct comparison here as a different TC set is being used and the original study used total travel time rather than a weighted combination of total travel time and work span. The current results imply that for typical cases where the number of jobs is much larger than the number of vehicles, the greatest benefit from CPFs is in the cases where there are at least some significant job-to-job differences in container configuration. This implies a slightly different take on some of the conclusions which came out of the original study. These results suggest that it would be important to include a variety of chassis types at the CPFs, and consider the inclusion of different types of chassis in the modeling process, as a topic for future research.
Figure 88. Scenario 1: Total cost improvement due to use of CPFs. In Scenario 1, a CPF is used only if the container configuration at origin does not match the container configuration at the destination of a job. The percent improvement is computed in comparison to the case when no CPFs are used.
Figure 89. Scenario 1: Solution (a) w/o CPFs and (b) w/ CPFs, \( N = 15 \), \( P = 1,200 \) seconds.
Figure 90. Scenario 1: Solution (a) w/o CPFs and (b) w/ CPFs, N = 60, P = 1,200 seconds.
Figure 91. Scenario 2: Total cost improvement due to use of CPFs.
In Scenario 2, the truck always uses a CPF when transitioning from one job to the next. The improvement is computed in comparison to the case when no CPFs are used.
Figure 92. Scenario 2: Solution (a) w/o CPFs and (b) w/ CPFs, N = 15, P = 1,200 seconds.
Figure 93. Scenario 2: Solution (a) w/o CPFs and (b) w/ CPFs, $N = 60$, $P = 1,200$ seconds.
3.3.3 Sensitivity Analysis for $\mu$

A sensitivity analysis was performed for varying values of the weight $\mu$ in the objective function. In Equation (88) the objective function minimizes the weighted sum of the total travel time needed to complete all vehicles’ schedules, $\sum_{m=1}^{M} T(s_m)$, and the maximum work span across all of the vehicles’ schedules, $\max_{m=1,\ldots,M} T(s_m)$. As the weight $\mu$ increases, it is anticipated that the objective function will force more uniformity across the vehicles’ schedules such that all vehicles will have approximately the same work span. This sensitivity analysis was performed using values of $\mu$ between zero and one, for $P = 1,200$ seconds and for a total number of jobs $N = 60$.

The total travel time with respect to the weight $\mu$ is shown in Figure 94. The maximum work span $\max_{m=1,\ldots,M} T(s_m)$ and the minimum work span $\min_{m=1,\ldots,M} T(s_m)$ are shown in Figure 95.

For $\mu = 0$, no optimization is performed with respect to the maximum work span, so that as seen in Figure 95, there is a large variation between the minimum and maximum work span of the drivers. The total travel time, however, is at its minimum when $\mu = 0$.

As $\mu$ increases the maximum and minimum work spans converge so that the drivers are more equally loaded as seen in Figure 95, and there is a slight increase observed in the total travel time (by less than 0.5%), shown in Figure 94. Depending upon the policies in place for driver pay (e.g. minimum hours to be worked per day, cost of overtime pay, etc.), $\mu$ could be tuned to allow for optimization of the overall cost to the TC. As the variation between the maximum and minimum work spans converges to a reasonable value of approximately 7 hours for $\mu = 1$, this value of $\mu$ was used for the nominal setting in the other case studies.
Figure 94. Total travel time vs. $\mu$.
The parameter $\mu$ is the weight used in the objective function as defined in Equation (88).

Figure 95. Work span vs. $\mu$.
The parameter $\mu$ is the weight used in the objective function as defined in Equation (88).
3.3.4 Impact of time-varying model route accuracy

In order to evaluate the benefits of including improved accuracy real-time travel duration predictions to allow for re-routing of vehicles during the day, simulation scenarios were run through the genetic algorithm with varying levels of travel duration prediction errors and with and without traffic accidents. The nominal values indicated in Section 3.2 were used for the case study and genetic algorithm settings, with the exception that \( P \) was fixed at 1,200 seconds, and the genetic algorithm generations were increased to 2,000 in order to provide increased accuracy in the results. First, the genetic algorithm was run with noise injected on the average data set using the baseline noise and noise scale factor as described in Section 3.2.2.3, resulting in an optimal schedule for each noise level. To assess the impact of route optimization using inaccurate data, the costs of the resultant schedules optimized with noisy data were calculated, using the true travel times in the objective function. Figure 96 below shows the percent degradation (increase) in the objective function when using a solution optimized with noisy data as compared to that optimized using the true travel times as a function of the noise scale factor. The resultant RMSE as a function of time for the four different noise cases is shown in Figure 97. In Figure 97 it can be seen that the peak RMSE evaluated for the largest noise scale factor was on the order of 20 minutes. The plot below shows the sensitivity of the optimal route solution to the noise in the predicted travel durations used for route planning. The impact of noisy data was assessed for Scenario 1 and Scenario 2 as defined in Section 3.3.2, where the impact of the noise for Scenario 1, which allows for direct routing between customer locations and MTs, is greater than that for Scenario 2. Across both cases and all noise scale factors, degradation of up to 7.5% was observed.
Figure 96. Percent degradation in solution due to errors in predicted travel durations. Percent degradation (increase) in objective function when using solution optimized with noisy data. In this figure: (i) noise scale factor is scaling of baseline noise envelope as described in Section 3.2.2.3; (ii) Scenario 1 and Scenario 2 as defined in Section 3.3.2; (iii) sixty jobs included; (iv) $P = 1,200$ seconds.
In addition to evaluating the impact of noise injection on the optimal routes, the impact of the accidents as defined in 3.2.2.5 was also evaluated. Fourteen different accident cases were evaluated, in each case injecting the accident at the same point in time, 8am, which corresponds to the first peak in the crash probability density from Figure 73. Each accident was injected as a two-hour additional delay on all routes impacted, and recovered over the period of two hours. No noise was injected, in order to best evaluate the impact of the accidents. However, none of these accident cases resulted in an impact on the solution, where comparing the “true” optimal solution provided by the genetic algorithm in the presence of the accident to that without the accident resulted in changes in the cost of less than 0.25%. In order to assess why this may have been the case the resultant RMSE as a function of time was once again plotted for the various noise cases as well as for all of the accident cases. Figure 98 shows the instantaneous
results over time for both the noise and accident cases, and Figure 99 shows the RMSE for the complete set of data (equivalent to the mean of the data in Figure 98) for the noise and accident cases. One can see with Figure 99 in particular that the noise levels for the individual accidents on average across the entire data set in Figure 99b were less than that in the 0.5 noise scale factor case in Figure 99a. Therefore, it is not surprising that the impact of a single accident on the overall route planning was not measurable, as it would be anticipated to have less impact than the smallest noise factor case. Since the individual accident cases as implemented herein did not have a measurable impact on optimization of the routes, the noise injection alone was used for the dynamic re-routing analysis which follows.

Figure 98. Route duration error magnitudes vs. time with accidents. Route duration error magnitudes are RMSE resulting from the comparison of noisy data to average data for all nodes used in case study at each point in time. In this figure: (i) noise scale factor is scaling of baseline noise envelope as described in Section 3.2.2.3; (ii) accidents injected as defined in Section 3.2.2.5 with 2 hour increased duration on effected routes.
Figure 99. Average route duration error magnitudes.
Route duration error magnitudes are RMSE resulting from the comparison of noisy data to average data for all nodes used in case study for the entire day. In this figure: (i) noise scale factor is scaling of baseline noise envelope as described in Section 3.2.2.3; (ii) accidents injected as defined in Section 3.2.2.5 with 2 hour increased duration on effected routes.

The potential benefits of a dynamic re-routing algorithm were evaluated using a case study and sensitivity analysis which provided bounds on improvements as a function of the increased accuracy in the real-time data. The dynamic re-routing software relies upon recursive calls to the genetic algorithm. These calls can be made at user selected time steps, where each sample includes a more accurate real-time data prediction for the current point of time. At each time step, the genetic algorithm is called such that any jobs which are in progress at the current point in time are fixed and cannot be altered by the genetic algorithm. The potential benefit of using dynamic re-routing depends upon the level of increased accuracy in the data used, where the three key items in assessing the benefit include: (i) the accuracy of the data used to make the original route determinations prior to that day; (ii) the accuracy of the real-time data which is queried; and (iii) the period over which the accuracy of the real-time data degrades to the original route prediction accuracies. For the analysis herein to provide an upper bound on the potential improvements, item (i) and (ii) were fixed with an assumed initial day route planning data accuracy and real-time data accuracy corresponding respectively to the five and 0.5 noise scale factor errors shown in
Figure 96, and Scenario 1 with direct routing between locations was assumed. Figure 100 shows the potential improvement as a function of leveraging the real-time data for a varying level of accuracy degradation periods. One can see that in the extremes once the period over which the accuracy degrades is long enough to cover the entire work span, the data converges to show the same relative benefit as would be expected between the noise scale factor errors at five and 0.5 from Figure 96, where the 7% improvement matches the anticipated improvement from the 7.5% degradation of the optimal solution for a noise scale factor of five to only a 0.5% degradation from the optimal solution for a noise scale factor of 0.5. On the other hand as the degradation period become too short, no improvement is afforded, as this is equivalent to repeatedly running the optimization algorithm with the same noisy data. Leveraging the work developed in order to model secondary accident rates as studied by (Moore II, Giuliano, & Cho, 2004) and (Sun & Chilukuri, 2010) a more realistic bound on the accuracy degradation window is likely on the order of two hours. For these more realistic accuracy degradation windows, there is potential for a small improvement of approximately 2% when using dynamic re-routing. However, it would be important to weigh the cost of the additional real-time queries needed against the potential benefits for the specific TC and job set in question prior to implementation.
Figure 100. Percent improvement in solution with real-time dynamic re-routing. Percent improvement (decrease) in objective function when using dynamic re-routing. In this figure: (i) initial day and real-time noise scale factors were 5 and 0.5 times baseline noise envelope; (ii) uses Scenario 1 as defined in Section 3.3.2; (iii) sixty jobs included; (iv) P = 1,200 seconds; (v) accuracy degradation period as defined in Section 3.2.2.6.

3.4 Summary

This second study examined the scheduling of chassis and container movements at the operational level, from the point of view of individual TCs, when CPFs are available for use, in the vicinity of a container port within a major metropolitan area. An optimization methodology was developed, which minimizes the travel time for the fleet of trucks used by the TC, and at the same time tries to minimize the variations in work load among individual drivers during the day. The optimal solution was obtained through the application of a genetic algorithm, developed specifically for this purpose. Time-varying dynamic models for the movements of chassis and containers were developed to be used in the optimization process. The effectiveness of the methodology was evaluated through a case study, focusing on the POLB, POLA and surrounding area. A location for the TC was chosen, and the company’s fleet of trucks was assigned a series of container movements to be completed during the day. Two scenarios were considered in the simulations: (1) when the truck does not have to pick up or drop off a chassis from one job to the next;
and (2) when the truck must pick up or drop off a chassis from the previous job to the next. Simulation results showed that the optimal solution obtained through the genetic algorithm provides improvements to the objective function when CPFs are used over the case when no CPFs are available.

When the number of jobs was small, the improvement was approximately 5% when Scenario 1 was applicable and as large as 30% when Scenario 2 is applicable. For large ratios of jobs to vehicles there was only a small benefit at approximately 1% for sixty jobs (allocated between ten vehicles) for Scenario 1, whereas Scenario 2 still had a larger benefit at up to 23%. From the point of view of a single TC (i.e. at the operational level), the benefit was similar to what was observed during the analysis at the strategic level in Section 2 which showed up to 21% improvements in total travel time for the overall network.

Note, however, that there is not a direct comparison here as a different TC set was used and the original study included total travel time rather than a weighted combination of total travel time and work span in the objective function. The current results imply that for typical cases where the number of jobs is much larger than the number of vehicles, the greatest benefit from CPFs is in the cases where there are at least some significant job-to-job differences in container configuration. This implies a slightly different take on some of the conclusions which came out of the original study. These results suggest that it would be important to include a variety of chassis types at the CPFs, and consider the inclusion of different types of chassis in the modeling process, as a topic for future research.

In addition to the formulation and optimization for initially planning daily activities, the study further modeled the problem in a dynamic environment, in which traffic network parameters can change drastically from initial daily predictions. In order to perform the optimization in a dynamic formulation with varying noise levels, a method by which noise could be injected into the initial daily predictions was developed as well as an incremental optimization approach. A modest potential benefit of approximately 2% may be expected if dynamic re-routing was performed. However, it would be important to weigh the cost of the additional real-time queries required against the potential benefits for the specific TC and job set in question prior to implementation.
4 CONCLUSIONS

This work studied the concept of CPFs and their potential impact on port drayage efficiency, including two major studies: one at the strategic planning level and one at the operational level for individual TCs.

In the first study, an analytical framework for modeling and optimization of chassis movements in transportation networks with CPFs was developed, and a case study in the LB/LA port area was performed. The results of this study indicate that a reduction of total travel time by up to 21% can be achieved when using the CPFs. The study also showed that, in the LB/LA port area, the return on investment for establishing additional CPF locations decreased sharply for any more than three CPFs.

In the second study, scheduling of chassis and container movements was optimized at the operational level for an individual TC. A multi-objective optimization problem was formulated in which the weighted combination of the total travel time for the schedules of all vehicles in the company fleet and the maximum work span across all vehicle drivers during the day was minimized. Time-varying dynamic models for the movements of chassis and containers were developed to be used in the optimization process. The optimal solution was obtained through a genetic algorithm, and the effectiveness of the developed methodology was evaluated through a case study which once again focused on the LB/LA port area. The case study used a TC located in the Los Angeles region, which utilized three candidate CPFs for exchange of chassis. The company assigned container movement tasks to its fleet of trucks, with warehouse locations spread across the region. In the simulation scenarios developed for the case study, the use of CPFs at the operational level provided improvements up to 30% (depending upon the specific scenario) over the cases of not using any CPFs. It was found in this work that for typical cases
where the number of jobs is much larger than the number of vehicles in the company fleet, the
greatest benefit from CPF use would be in the cases where there are some significant job-to-job
differences with respect to chassis usage.

Lastly, in addition to the formulation and optimization for initially planning daily activities, the
study further modeled the problem in a dynamic environment, in which traffic network
parameters can change drastically from initial daily predictions. In order to perform the
optimization in a dynamic formulation with varying noise levels, a method by which noise could
be injected to the initial daily predictions was developed as well as an incremental optimization
approach. A modest potential benefit approximately 2% may be expected if dynamic re-routing
was performed. However, it would be important to weigh the cost of the additional real-time
queries required for implementation against the potential benefits for the specific company and
job set in question prior to implementation.

Overall, at both the strategic and operational level, the findings indicate that travel time can be
significantly reduced through implementation of CPFs which has important implications in
reducing negative environmental impacts of the port as well as operational costs for trucking
companies.
5 REFERENCES


