

Rochester Institute of Technology

## RIT Digital Institutional Repository

---

Theses

---

2-17-2013

### Route-based transportation network design

Lifan Zhang

Follow this and additional works at: <https://repository.rit.edu/theses>

---

#### Recommended Citation

Zhang, Lifan, "Route-based transportation network design" (2013). Thesis. Rochester Institute of Technology. Accessed from

This Thesis is brought to you for free and open access by the RIT Libraries. For more information, please contact [repository@rit.edu](mailto:repository@rit.edu).

**Rochester Institute of Technology**

**ROUTE-BASED TRANSPORTATION NETWORK DESIGN**

**A Thesis**

**Submitted in partial fulfillment of the  
requirements for the degree of  
Master of Science in Industrial Engineering**

**in the**

**Department of Industrial & Systems Engineering  
Kate Gleason College of Engineering**

**by**

**Lifan Zhang**

**February 17, 2013**

DEPARTMENT OF INDUSTRIAL AND SYSTEMS ENGINEERING  
KATE GLEASON COLLEGE OF ENGINEERING  
ROCHESTER INSTITUTE OF TECHNOLOGY  
ROCHESTER, NEW YORK

CERTIFICATE OF APPROVAL

---

M.S. DEGREE THESIS

---

The M.S. Degree Thesis of Lifan Zhang  
has been examined and approved by the  
thesis committee as satisfactory for the  
thesis requirement for the  
Master of Science degree

Approved by

---

Dr. Mike Hewitt, Thesis Advisor

---

Dr. Ruben Proaño

## **ABSTRACT**

Given shipment demand and driving regulations, a consolidation carrier has to make decisions on how to route both shipments and drivers at minimal cost. The traditional way to formulate and solve these problems is through the use of two-step models. This thesis presents a heuristic algorithm to solve an integrated model that can provide superior solutions. The algorithm combines a slope scaling initialization phase and tabu search to find high-quality solutions. The performance of the proposed heuristic is benchmarked against a commercial solver and these results indicate that the proposed method is able to produce better quality solutions for the similar solution time.

## **ACKNOWLEDGEMENTS**

This thesis and my master degree would not have been possible without the tremendous help from several individuals.

First of all, my deepest gratitude goes to Dr. Mike Hewitt who has provided dedicated and continuous guidance throughout my study at RIT and my thesis research. His knowledge and insights have been the most inspiring source of ideas for my experiments and analysis. With his endless patience, trust, encouragement, and his humor that has made my journey of research more fun and less stressful, I have felt very lucky and grateful to be able to work with him.

I would like to thank Dr. Ruben Proaño for his knowledgeable input to my research and his critical advices on my document, all of which are indispensable for the completion of this thesis.

My sincere thanks go to my friends and classmates at RIT for their wonderful companion and support made my time here even more enjoyable.

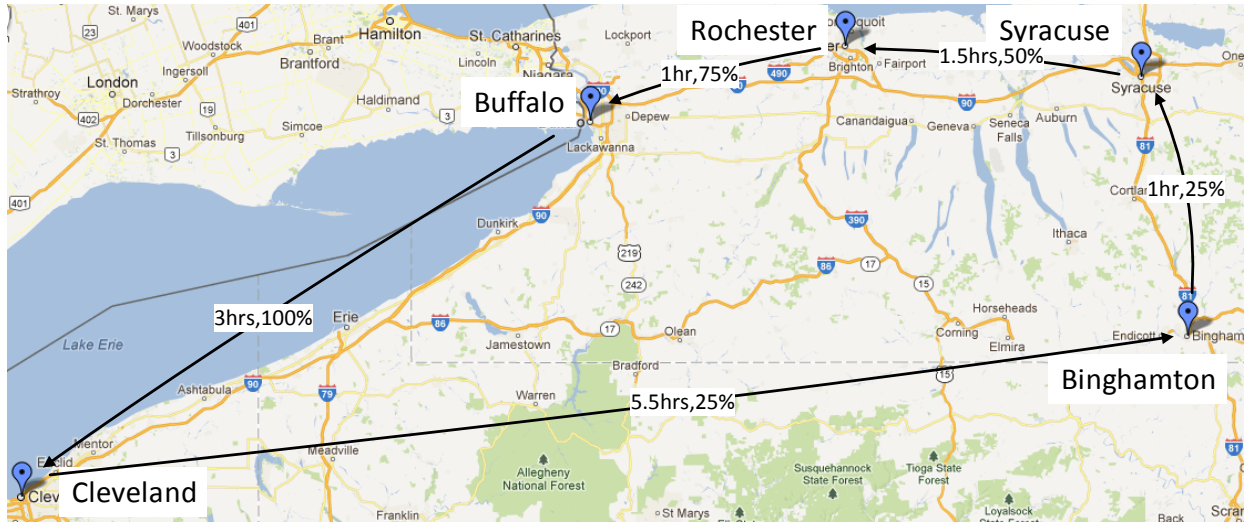
Last but not least, I am indebted to my family back home, especially my parents, for their indefinite support, both spiritually and materially. Words cannot describe my gratefulness enough; therefore I dedicate this thesis to them.

## Table of Contents

1	INTRODUCTION .....	1
2	PROBLEM STATEMENT .....	7
3	LITERATURE REVIEW .....	11
4	ALGORITHM PHASE ONE - SLOPE SCALING INITIALIZATION .....	13
4.1	Slope Scaling Procedure .....	15
4.2	Routes Generation using DFS.....	19
5	ALGORITHM PHASE TWO - TABU SEARCH.....	22
5.1	Route-based Neighborhood.....	24
5.2	Tabu Search .....	27
5.3	Route Set Expansion .....	31
6	RESULTS AND ANALYSIS.....	33
7	CONCLUSIONS AND FUTURE WORK .....	38
	REFERENCES .....	40
	APPENDICES – DETAILED RESULTS .....	43
	APPENDIX A.....	43
	APPENDIX B .....	44

# 1 INTRODUCTION

The trucking industry is vital to US economic prosperity and it represents about 5% of gross domestic product ([2]). In 2007, about \$8 trillion of merchandise, or 71% of all freight transported in US was delivered by trucks ([5]). The industry is not only enormous in size, but also diverse and complex and carriers can be classified into truckload or consolidation. A truckload carrier picks up a full load of shipment and delivers directly to the destination. In this case, the carrier provides tailored services because it assigns a truck solely to one customer and the loading/unloading is performed at locations designated by the customer. On the other hand, consolidation carriers transport shipments that do not fill up one container and they are the focus of this thesis. For them, it is economically unfavorable to send a shipment directly from origin to destination. To increase truck utilization, a carrier consolidates shipments from multiple shippers and routes them through a transportation network consisting of consolidations terminals where shipments are further sorted, loaded and unloaded. Therefore the shipments are often routed on a path other than the shortest between origin and destination.



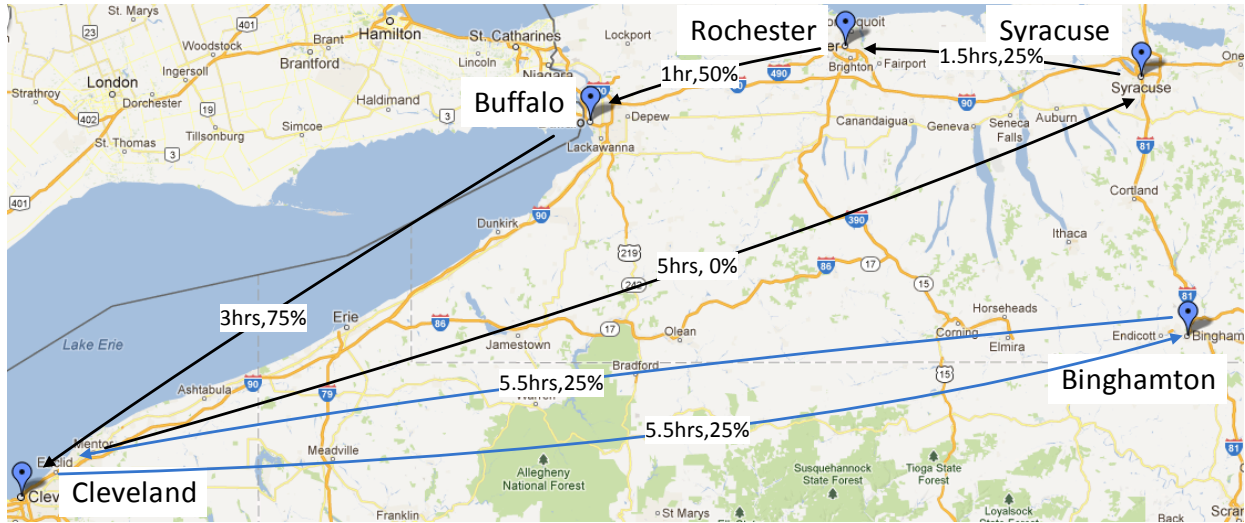
**Figure 1-1**

To illustrate consolidation, consider Figure 1-1, suppose there are shipments originating in Binghamton, Syracuse, Rochester and Buffalo, and all are destined for Cleveland and another shipment originating in Cleveland must be delivered to Binghamton. Also assume that each shipment fills 25% of truck capacity. By routing through

Binghamton→Syracuse→Rochester→Buffalo→Cleveland

to pick up, consolidate and unload shipments at Cleveland, the carrier can reach truck utilization of 25%, 50%, 75% and 100% respectively on each segment of the route, as denoted besides the links between cities in Figure 1-1. Then the carrier picks up the last shipment and returns to Binghamton. In the alternative scenario of delivering shipments directly between an origin and a destination, another truck has to be added to route the shipments directly between Binghamton and Cleveland with utilization of 25% on each way, as shown in Figure 1-2.



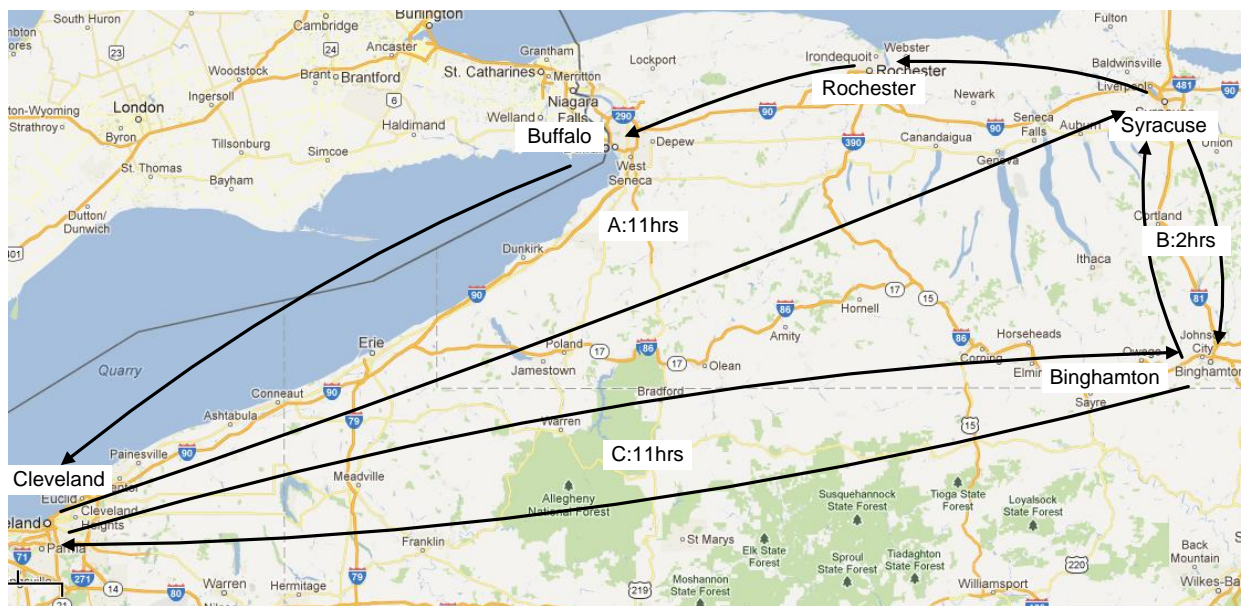


**Figure 1-2**

To ensure traffic safety and drivers health, many laws and business rules related to commercial driving are proposed by different sectors such as federal and state government, labor union and trucking companies themselves. One that places a great impact on motor carriers' behavior is Hours-of-Service (HOS) Regulations issued by the Department of Transportation. HOS, comprised of a set of laws on drivers' daily and weekly working hours, intend to reduce fatigue-related traffic accidents and drivers' chronic health problems ([31]). As an example, according to HOS, a property-carrying commercial driver may drive a maximum of 11 hours after 10 consecutive hours off duty. As a result, the route in Figure 1-1 with travelling time of 12 hours violates this rule and would become ineligible for carriers. Another constraint resulting from union contracts is that a driver must return to his/her home at the end of a workday. In this case, a driver's travelling has to start and end at the same location.

It is estimated is that 48% of total spending of trucking companies accounts for fuel, maintenance and tolls ([1]). Since these costs are essentially proportional to travel distance/time, it has become one of the primary concerns to design a consolidation network that satisfies

business demand and complies with the regulations mentioned above while minimizing operating cost. The traditional approach of network design is to form and solve the problem through two stages ([24]). Stage one is a variant of the capacitated multi-commodity network design problem (CMND). Given data about shipments and cost, and assuming resources (drivers) are always available, CMND selects links between cities to install in the network and routes shipments accordingly. Figure 1-1 in fact illustrates the solution to the stage one problem. Stage two is a vehicle routing problem (VRP), which routes necessary resources to transport the shipments on paths prescribed by the stage one solution while taking driving regulations as constraints. Figure 1-3 shows the 3 routes in the solution to the stage two model.



**Figure 1-3**

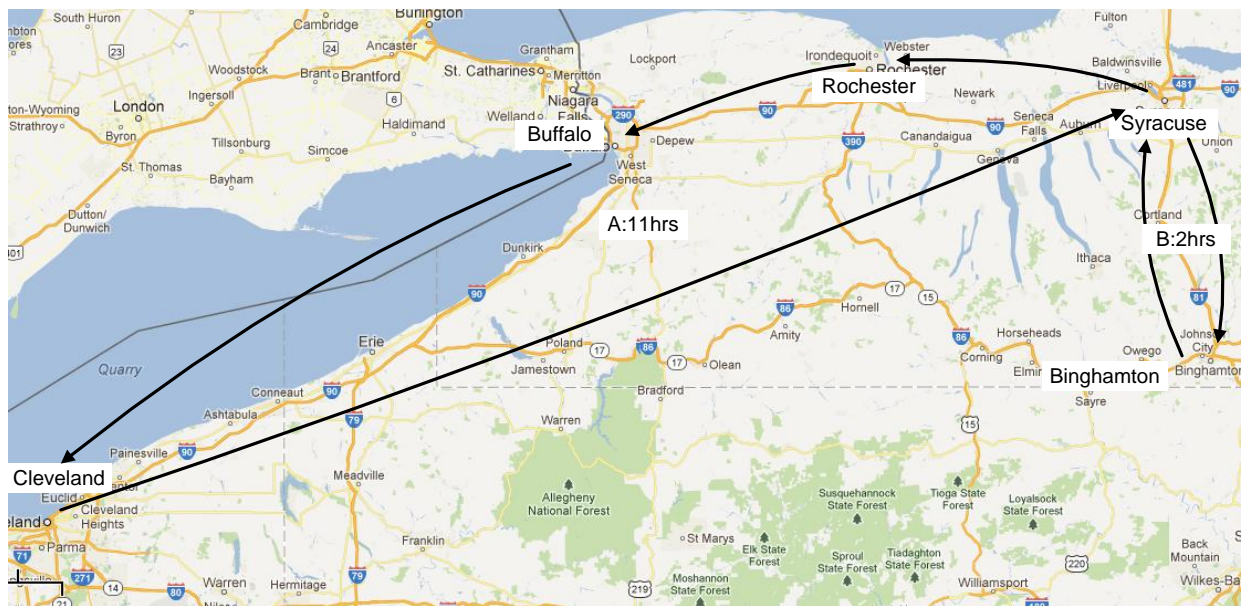
As seen in Figure 1-3, such a two-stage solution includes following routes:

Route A Syracuse→Rochester→Buffalo→Cleveland→Syracuse, 11 hours

Route B Binghamton→Syracuse→Binghamton, 2 hours

Route C Binghamton→Cleveland→Binghamton, 11 hours.

The total travel time is 24 hours, however the solution is not optimal because the shipment from Cleveland can be delivered by Route A going back to Syracuse and Route B going back to Binghamton, respectively. Therefore Route C between Cleveland and Binghamton is not necessary anymore, yielding 2 routes and total time of 13 hours, shown in Figure 1-4.



**Figure 1-4**

As a matter of fact, if one jointly solves the two problems using an integrated model (IM), one will obtain this better solution. Unlike the two-stage approach that moves commodities based on individual links between two locations, the integrated model considers routes, which are made up of a set of connecting links. In another word, a route can be seen as a sequence of different places to visit. In this example, available routes include Route A, B, C, and other paths that start and end at the same city. Considering the high turnover rate and shortage of truck drivers that have increased the freight rates continually ([16]), another merit of the integrated model is that it allows carriers to set up drivers' preferred routes and potentially helps companies creating a steady workforce. The development of algorithms for the integrated model has been limited; and

this thesis aims to fill this void. Therefore the focus of the thesis is to develop an efficient algorithm for solving the integrated model, which can produce high quality solutions (i.e., a low cost network) in a timely manner.

The following thesis is organized as follows. Chapter 2 introduces the formulation of the problem. Chapter 3 reviews the relevant literature. The proposed methodology is described in Chapter 4 and 5. Computational results are presented and analyzed in Chapter 6. Chapter 7 presents the conclusions and future work.

## 2 PROBLEM STATEMENT

Hewitt [24] performed computational experiments to compare the quality of solutions to the IM and the two-stage models on small instances and found that the IM yields solutions that are approximately 20% better. However the time needed for solving IM is significantly longer, therefore the purpose of this thesis is to design an efficient and effective heuristic for the integrated model. The IM in essence combines the following two models into one, and we first introduce the two-stage here.

The model solved in first stage takes the form of CMND, in which a given set of commodities are routed on a directed network. The commodities have their shipment origin, destination, and quantity/demand respectively. The network is built on a directed graph made of nodes and capacitated arcs, and the total cost is associated to the final selection of arcs.

The graph is denoted as  $G = \{\mathcal{N}, \mathcal{A}\}$ , with  $\mathcal{N}$  being the set of nodes and  $\mathcal{A}$  the set of arcs. In  $\mathcal{A}$ , let  $(i, j)$  represent an arc from node  $i$  to node  $j$  with fixed cost  $f_{ij}$ , variable cost  $c_{ij}$  and capacity  $u_{ij}$ . Associated with each commodity  $k$  is a source node  $s(k)$ , sink node  $t(k)$  and demand quantity  $d^k$ . Let variable  $x_{ij}^k$  denote the fraction of the volume of commodity  $k$  that is routed on arc  $(i, j)$  and let  $y_{ij}$  be the number of times that arc  $(i, j)$  is used. The resulting formulation is

$$\min \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{A}} c_{ij} (d^k x_{ij}^k) + \sum_{(i,j) \in \mathcal{A}} f_{ij} y_{ij}$$

subject to

$$\sum_{j:(i,j) \in \mathcal{A}} x_{ij}^k - \sum_{j:(j,i) \in \mathcal{A}} x_{ji}^k = \delta_i^k, \quad \forall i \in \mathcal{N}, \forall k \in \mathcal{K} \quad (1)$$

$$\sum_{k \in \mathcal{K}} d^k x_{ij}^k \leq u_{ij} y_{ij}, \quad \forall (i,j) \in \mathcal{A} \quad (2)$$

$$y_{ij} \in \mathbb{Z}_+, \quad \forall (i,j) \in \mathcal{A} \quad (3)$$

$$0 \leq x_{ij}^k \leq 1, \quad \forall k \in \mathcal{K}, \forall (i,j) \in \mathcal{A} \quad (4)$$

Constraints (1) represents flow balance, where  $\delta_i^k$  is 1,0 or -1 when node  $i$  is the source, sink, or intermediate node for the distribution of commodity  $k$ , respectively. Specifically, the total flow of the commodity that leaves its source node or enters into its sink node has to be equal to commodity's demand. For any intermediate node, the flow entering into it is equal to the flow leaving it. Constraints (2) ensures that there is enough capacity for the flow on the arc, and that the arc has to be installed (fixed cost paid) before it can be used. In most applications of CMND, design variables  $y_{ij}$  are restricted to 0 or 1, where  $y_{ij} = 1$  when arc  $(i,j)$  is selected in the network and  $y_{ij} = 0$  when arc is closed. While in our problem,  $y_{ij}$  takes integer because it represents the number of units of capacity installed. In (4), we allow commodities to travel on multiple paths from their origin to destination.

With solution  $(\bar{x}, \bar{y})$  to the stage one model, driving resources are needed to execute the prescribed shipment routes. Therefore, the stage two model is solved to determine the actual routing plan and often referred as an arc covering problem ([24]). Let  $b_{ij} = \sum_{k \in \mathcal{K}} d^k \bar{x}_{ij}^k$ , be the amount of commodities assigned to arc  $(i,j)$ , and let  $\theta$  be a set of valid resource routes on  $G$  that contain arcs from the solution  $(\bar{x}, \bar{y})$ . We assume that each route has fixed cost  $f_\tau = \sum_{(i,j) \in \tau} f_{ij}$  and capacity  $u_\tau$ . The definition of a valid route depends on the regulations considered. For

example, if each arc has travel time  $t_{ij}$  hours, and a driver can drive up to 11 hours, then  $\theta$  will only contain routes  $\tau$  such that  $\sum_{(i,j) \in \tau} t_{ij} \leq 11$ . Note that assuming that a route's fixed cost is proportional to its travel time, the requirement of a valid route  $\tau$  can be written as  $\sum_{(i,j) \in \tau} f_{ij} \leq \Delta$ , where  $\Delta$  is the equivalent upper limit of route cost. Given  $\theta = \{\tau: \sum_{(i,j) \in \tau} f_{ij} \leq \Delta\}$ , the stage two model becomes

$$\min \sum_{\tau \in \theta} f_{\tau} z_{\tau}$$

subject to

$$b_{ij} \leq \sum_{\tau \in \theta: (i,j) \in \tau} u_{\tau} z_{\tau}, \quad \forall (i,j) \in \mathcal{A} \quad (5)$$

$$z_{\tau} \in Z_{+}, \quad \forall \tau \in \theta$$

Here  $z_{\tau}$  indicates the number of times the resource route  $\tau$  is used or the number of copies of the route that are installed and the objective is to minimize the sum of selected route cost. Constraint (5) ensures that enough routes are set up to ship commodities on arc  $(i,j)$ .

In the integrated model (IM), the problem definitions and constraints are combined to yield the following formulation.

$$\min \sum_{\tau \in \theta} f_{\tau} z_{\tau}$$

subject to

$$\sum_{j: (i,j) \in \mathcal{A}} x_{ij}^k - \sum_{j: (i,j) \in \mathcal{A}} x_{ji}^k = \delta_i^k, \quad \forall i \in \mathcal{N}, \forall k \in \mathcal{K} \quad (1)$$

$$\sum_{k \in \mathcal{K}} d^k x_{(i,j,\tau)}^k \leq u_\tau z_\tau, \quad \forall (i,j) \in \tau, \forall \tau \in \theta \quad (6)$$

$$\sum_{\tau \in \theta} x_{(i,j,\tau)}^k = x_{ij}^k, \quad \forall (i,j) \in \mathcal{A} \quad (7)$$

$$z_\tau \in Z_+, \quad \forall \tau \in \theta$$

$$0 \leq x_{ij}^k \leq 1, \quad \forall k \in \mathcal{K}, \forall (i,j) \in \mathcal{A}$$

$x_{(i,j,\tau)}^k$  is the percentage of commodity  $k$  travelled on arc  $(i,j)$  of route  $\tau$ , which means an arc can be served by different routes. Constraint (6) makes sure that the amount of flow on the arc of a certain route cannot exceed the capacity provided by the route.



### 3 LITERATURE REVIEW

Planning decisions in freight transportation management occurs at three levels ([9]). Strategic level decisions determine the general physical network and operation strategies within international, national or regional scope. Tactical level planning involves service network design that designates the type and schedule of transport services, as well as traffic routes, in order to achieve efficient allocation and utilization of resources. Operational level planning includes the actual implementation of scheduled services and all dynamic activities that are performed by local management.

On a tactical level, network design models are widely used to assist in making decisions regarding logistics infrastructure and service network. The CMND is a particular case of network design and has been extensively studied and identified as one of the most challenging optimization problems to solve ([8]), and interested readers can refer to Gendron et al [14] and Magnanti and Wong [28] for a survey of models and algorithms of this problem. For solving CMND, exact algorithms are impractical for instances of realistic size; thus, various heuristics approaches have been developed. For example, a tabu search procedure combining simplex pivoting and column generation was introduced by Crainic et al [10]. Ghamlouch et al [15] constructed a cycle-based neighborhood for a tabu search algorithm by generating cycles on which commodities can be rerouted. An approach taking advantage of both exact and heuristic algorithms was developed by Hewitt et al [25]. Powell [30] presented a local heuristics with add and drop procedures and Jarrah et al [26] proposed a freight path tree generation approach.

A service network design problem optimizes routes and schedules with regard to level of service and resource constraints. Pederson et al [29] addressed service network design with

resources requirements, such as an equal number of service resource leaving and entering every terminal in the network. Models in Erera et al [12] captured empty trailer movements and timing of consolidation. One of the few exact algorithms designed for a service network problem can be found in Anderson et al [3]. Anderson et al [4] presented studies and comparisons of four alternative formulations of service network problem with assets management criteria.

For the vehicle routing problem (VRP) which minimizes routing cost with respect to nodes/arcs visiting constraints, Goel [18] provided two methods for problems considering driver rules in the European Union. Feillet et al [13] proposed a branch-and-price algorithm for solving a resource routing problem with profits and constraints including maximal length of routes. Surveys of arc routing problem can be found in Hertz [21] and Haouari [20]. A tabu search procedure for an arc routing problem with constraints of arcs requirements and vehicles capacity was introduced in Hertz et al [22]. A tabu search heuristic for vehicle routing and driving scheduling problem with regard to US regulation was introduced in Rancourt et al [30]. Recently Goel and Vidal [19] studied a similar problem considering international service regulations and presented a hybrid genetic algorithm that combines local search and driver schedule generation.

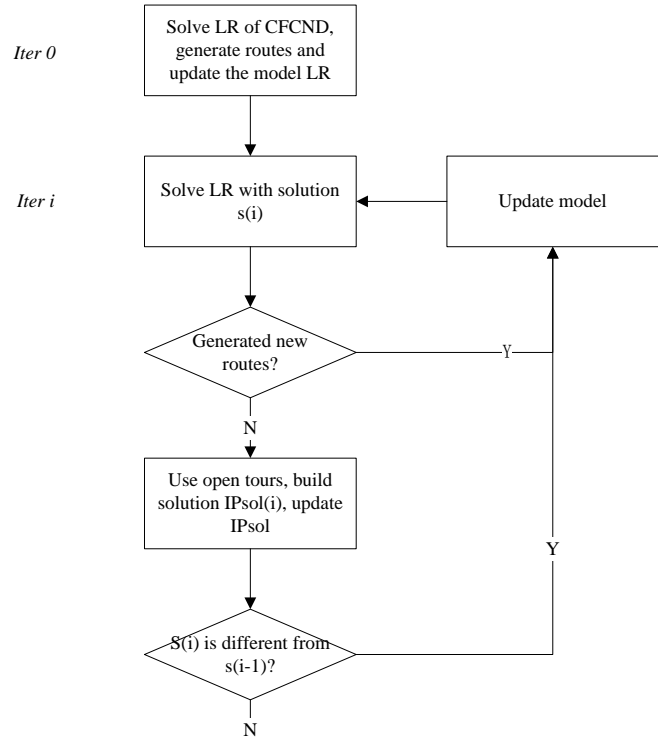
Despite the number of studies addressing each type of problem individually, research that deals with the CMND and the VRP jointly is limited and this thesis aims to fill the gap.

## **4 ALGORITHM PHASE ONE - SLOPE SCALING INITIALIZATION**

In this study, we developed a tabu-search based heuristic algorithm because for optimization problems of large size, heuristics have outperformed exact algorithm in terms of time efficiency and solution quality ([15]). Instead of spending a prohibitively long time to perform an exhaustive search and find the optimal solution, heuristics provide satisfactory solutions in a shorter period of time without enumerating all solutions.

The entire proposed heuristic is illustrated in the flow chart (Figure 4-1). The first phase provides initial solution by adapting slope scaling technique and generating routes at the same time and will be presented in this chapter. The second phase finds better solution by selecting neighbor solutions using tabu search and will be presented in Chapter 5.

## Slope Scaling Initialization (Chapter 4)



## Tabu Search (Chapter 5)

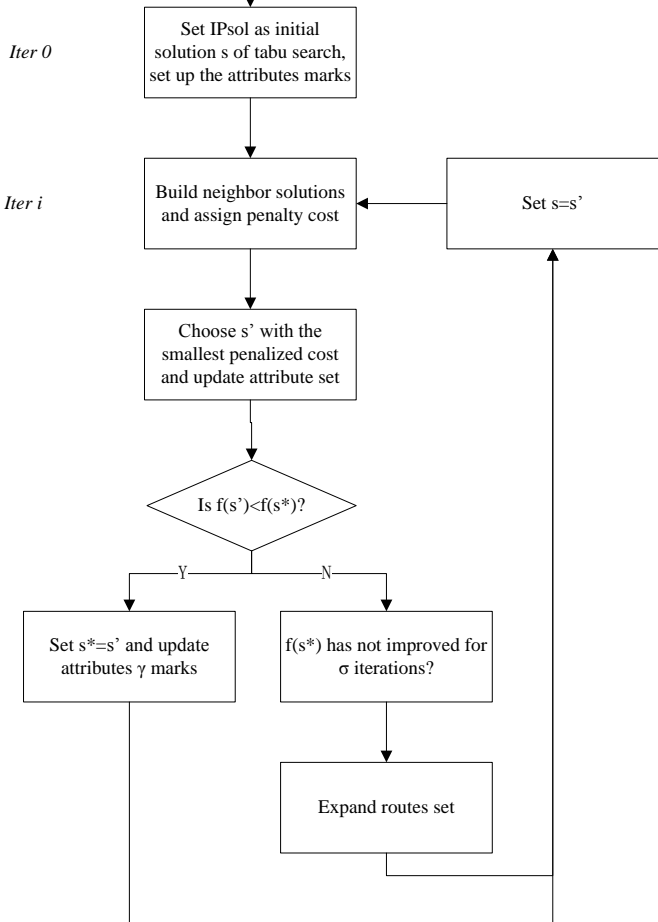


Figure 4-1

#### 4.1 Slope Scaling Procedure

In our problem, an initial solution is required for the heuristic search to work, and the route set  $\theta$  has to be generated before solving IM. However in real-world instances, enumerating all possible routes upfront will take too much time. Therefore, routes that we anticipate will appear in high quality solutions are generated dynamically. We adopt the technique of slope scaling (SS), developed by Kim et al [27] to produce initial solutions to IM, and add a step to generate routes at the same time. In every iteration, SS solves a linear approximation of IM and updates the linear cost coefficients in the objective function to reflect the actual cost incurred by the solution ([11]). The process iterates until the objective function value of two successive solutions are identical.

To start, the SS in iteration 0 solves the linear relaxation (LR) of the CMND and collects the open arcs in the solution.

*Iteration 0:*

Solve

$$\min \sum_{(i,j)} f_{ij} y_{ij}$$

subject to

$$\sum_j x_{ij}^k - \sum_j x_{ji}^k = \delta_i^k, \quad \forall i \in \mathcal{N}, \forall k \in \mathcal{K} \quad (1)$$

$$\sum_{k \in \mathcal{K}} d^k x_{ij}^k \leq u_{ij} y_{ij}, \quad \forall (i,j) \in \mathcal{A} \quad (2)$$

$$0 \leq x_{ij}^k \leq 1, \quad \forall (i,j) \in \mathcal{A}$$

$$y_{ij} \geq 0, \quad \forall (i,j) \in \mathcal{A}$$

The optimal solution is denoted as  $(\hat{x}_{ij}^{k(0)}, \hat{y}_{ij}^{(0)})$ . All open arcs, or arcs such that  $\hat{y}_{ij}^{(0)} > 0$ , are added to  $\mathcal{A}^0$ , which corresponds to the set of open arcs that need to be covered by routes and thus drive the generation of new routes.. To illustrate, Figure 4-2(a) shows a CMND with shipment demand denoted beside the origin node. Suppose solid lines in Figure 4-2(b) represents the optimal solution to its LR, in which  $y_{12}, y_{23}, y_{35}, y_{34} > 0$  and therefore  $\mathcal{A}^0 = \{(1,2), (2,3), (3,5), (3,4)\}$ .

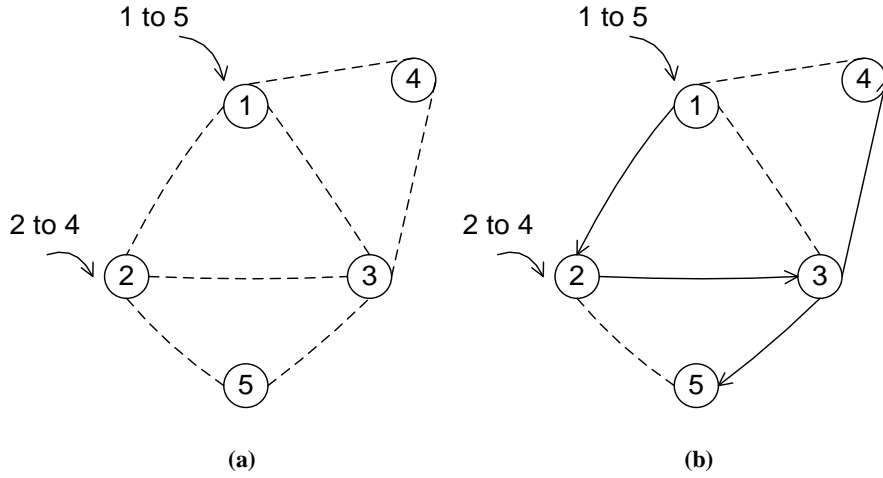


Figure 4-2

Iteration  $n$  ( $n > 0$ ):

Routes covering arcs in  $\mathcal{A}^0$  are generated using depth-first search (DFS) and added to  $\theta$ .

The detailed procedure of DFS is introduced in Section 4.2. With the optimal solution  $(\hat{x}_{ij}^{k(n-1)}, \hat{y}_{ij}^{(n-1)}, \hat{z}_{\tau}^{(n-1)})$  in iteration  $(n-1)$ , the routes fixed cost  $\bar{f}_{\tau}^{(n)}$  in iteration  $n$  is adjusted to reflect the actual cost of the routes needed to carry the current flow. Since  $f_{\tau}$  is made of the cost of each

arc on the route, with  $m_{ij}^n$  defined as the arc adjusted cost,  $\bar{f}_\tau^{(n)} = \sum_{(i,j) \in \tau} m_{ij}^n$ . And  $m_{ij}^n$  is calculated to satisfy  $\sum_k d^k \hat{x}_{ij}^{k(n-1)} m_{ij}^n = f_{ij}$  when  $\sum_k d^k \hat{x}_{ij}^{k(n-1)} > 0$ , or  $m_{ij}^n$  equal to the original arc cost  $f_{ij}$  when  $\sum_k d^k \hat{x}_{ij}^{k(n-1)} = 0$ . These conditions further lead to the update of  $m_{ij}^n$  in iteration  $n$  as follows.

$$\begin{cases} m_{ij}^n = \frac{f_{ij}}{\sum_k d^k \hat{x}_{ij}^{k(n-1)}}, \text{ if } \sum_k d^k \hat{x}_{ij}^{k(n-1)} > 0 \\ m_{ij}^n = f_{ij}, \text{ if } \sum_k d^k \hat{x}_{ij}^{k(n-1)} = 0 \end{cases}$$

The motivation is that, if the flow in the current solution is too small to justify the fixed cost of the route, the high linearized cost  $\bar{f}_\tau^{(n)}$  should encourage the optimization in the next iteration to either increase the flow so that the marginal fixed cost is decreased, or drop the flow to zero so that there is no cost associated with the route.

Note that starting from iteration  $n \geq 2$ , the cost coefficients of routes that existed in the previous iteration are updated as follows:

$$\begin{cases} \text{If } \hat{z}_\tau^{(n-1)} > 0, & \bar{f}_\tau^{(n)} = \sum_{(i,j) \in \tau} m_{ij}^n \\ \text{If } \hat{z}_\tau^{(n-1)} = 0, & \bar{f}_\tau^{(n)} = \bar{f}_\tau^{(n-1)} \end{cases}$$

The updated model is

$$\min \sum_{(i,j) \in \mathcal{A} \setminus \mathcal{A}^0} f_{ij} y_{ij} + \sum_{\tau \in \Theta} \bar{f}_\tau^{(n)} z_\tau$$

subject to

$$\sum_{k \in \mathcal{K}} d^k x_{ij}^k \leq u_{ij} y_{ij}, \quad \forall (i, j) \in \mathcal{A} \setminus \mathcal{A}^0 \quad (9)$$

$$\sum_{k \in \mathcal{K}} d^k x_{ij}^k \leq \sum_{\tau \in \theta: (i, j) \in \tau} u_{\tau} z_{\tau}, \quad \forall (i, j) \in \mathcal{A}^0 \quad (10)$$

$$\sum_j x_{ij}^k - \sum_j x_{ji}^k = \delta_i^k, \quad \forall i \in \mathcal{N}, \forall k \in \mathcal{K} \quad (1)$$

$$0 \leq x_{ij}^k \leq 1, \quad \forall (i, j) \in \mathcal{A}$$

$$y_{ij} \geq 0, \quad \forall (i, j) \in \mathcal{A}$$

$$z_{\tau} \geq 0, \quad \forall \tau \in \theta$$

Here, arcs  $(i, j) \in \mathcal{A} \setminus \mathcal{A}^0$  are not covered by any route yet (we refer them as isolated arcs). Arc capacity constraints set (2) are divided into two types based on whether the arc is isolated (9) or not (10). Original flow balance constraints (1) remain the same, as do the variable definitions.

The updated model is solved and its optimal solution is defined as  $(\widehat{x}_{ij}^{(n)}, \widehat{y}_{ij}^{(n)}, \widehat{z}_{\tau}^{(n)})$ .

And we again collect open isolated arcs and update  $\mathcal{A}^0$ , formally,

$$\begin{cases} \text{If } \widehat{y}_{ij}^{(n-1)} > 0, f_{ij} = 0, \mathcal{A}^0 = \mathcal{A}^0 \cup (i, j) \\ \text{If } \widehat{y}_{ij}^{(n-1)} = 0, \text{ do nothing} \end{cases}$$

After every iteration  $n$  in which all  $\widehat{y}_{ij}^{(n)} = 0$ , indicating that no isolated arc is carrying flow and all flow are assigned to some routes, the IM is solved with only these open routes, which means to take  $\theta' = \{\tau: \widehat{z}_{\tau}^{(n)} > 0\}$  as the set of available routes. The solution to this sub-



model is referred as  $IPsol^n$ , which is a feasible solution to the original problem. Different feasible solutions will be produced and collected in this phase. The procedure stops when there is no further modification in the objective function value from one iteration to the next, as well as no more new routes are generated. From the collected set of  $IPsol^n$ , the one with best solution value is chosen to be the initial solution  $IPsol$  for tabu search.

## 4.2 Routes Generation using DFS

DFS is an order of visiting nodes or searching arcs in a graph ([7]). To start, it selects an arbitrary arc  $(i, j_1)$  from the graph, labels node  $i$  and  $j_1$  as ‘visited’, then traverses another arc  $(j_1, j_2)$ . If  $j_2$  is visited, the search goes back to  $j_1$  and chooses a different arc; otherwise it labels  $j_2$  as visited and takes it as current node and repeats the above procedure. When the search reaches the starting node  $i$  through visiting arcs  $(i, j_1), (j_1, j_2) \cdots (j_m, i)$ , a route consisting of these arcs is generated and added to  $\theta$ . Notice that for our experiments, different starting nodes do not differentiate a route. It is to say that route  $1 \rightarrow 2 \rightarrow 3 \rightarrow 1$  or  $2 \rightarrow 3 \rightarrow 1 \rightarrow 2$  etc are all stored as one single route that covers  $(1,2), (2,3)$  and  $(3,1)$ . However our method can be adapted to applications where the starting node matters, such as when there are a fixed number of drivers/vehicles at a terminal.

Similarly, the DFS can be executed with different constraints on a route, such as number of arcs or total travel time. In our application, each arc has a certain units of travel time/length, and a valid route’s maximal driving hours according to the regulation is simplified as a maximum value such that DFS keeps adding the travel time of visited arcs and forms a route only when it does not exceed the maximum value. The pseudocode of DFS procedure on a graph with starting point of  $i$  is presented as follows.

Set  $i^* = i$

DFS ( $graph, i, i^*$ )

Label  $i$  as visited

**For** ( $i, j$ ) in  $outbound(i)$  **do**

**If**  $j$  is not visited and total travel time +  $t_{ij} \leq \text{limit}$  **then**

Label  $j$  as visited

Total travel time = total travel time +  $t_{ij}$

Call DFS ( $graph, j, i^*$ )

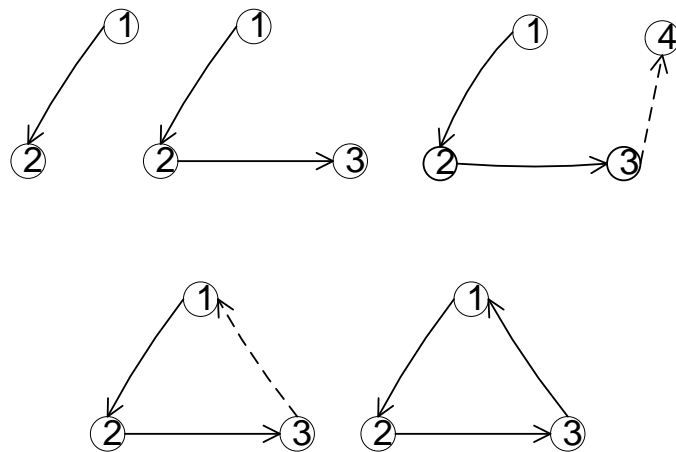
**Else if**  $j = i^*$  and total travel time +  $t_{ij} \leq \text{limit}$  **then**

Stop

**Else**

Continue

Taking arc set  $\mathcal{A}^0 = \{(1,2), (2,3), (3,5), (3,4)\}$  for example, suppose the total travelling time of a route must not exceed 3 units and each arc has traveling time of 1 unit. In Figure 4-3, depth first search starts at node 1, traverses along arc (1,2) updating total time to 1 unit, then (2,3) updating time to 2 units, then (3,4) with maximum time allowed of 3 units, but the search does not arrive at starting node. The procedure goes back to the most recently visited node, which is 3, takes (3,1) and a route  $1 \rightarrow 2 \rightarrow 3 \rightarrow 1$  is formed.



**Figure 4-3**

## 5 ALGORITHM PHASE TWO - TABU SEARCH

Tabu search is a type of neighborhood search scheme that constructs a set of solutions  $N(s)$  obtained by applying a set of moves to a current solution  $s$ . Examples of moves are changing a variable value, adding or deleting some elements from a set, etc ([17]). Then the procedure selects the next solution  $s'$  from  $N(s)$  and decides whether to stop or repeat the process from  $s'$ . Most neighborhood search methods keep in memory the best solution ever found and compare it with candidate solutions during the exploration. With the set of moves associated with current solution  $s$  defined as  $D(s)$ , and applying a move  $d$  to  $s$  as  $s \oplus d$ , the formal definition of the search starting from a solution  $s$  is below.

SEARCH( $s$ )

**For**  $d \in D(s)$  **do**

$s' = s \oplus d, N(s) = N(s) \cup s'$

Choose  $\hat{s}$  from  $N(s)$

Set  $s = \hat{s}$

**If**  $f(\hat{s}) < f(s^*)$ , **do**:

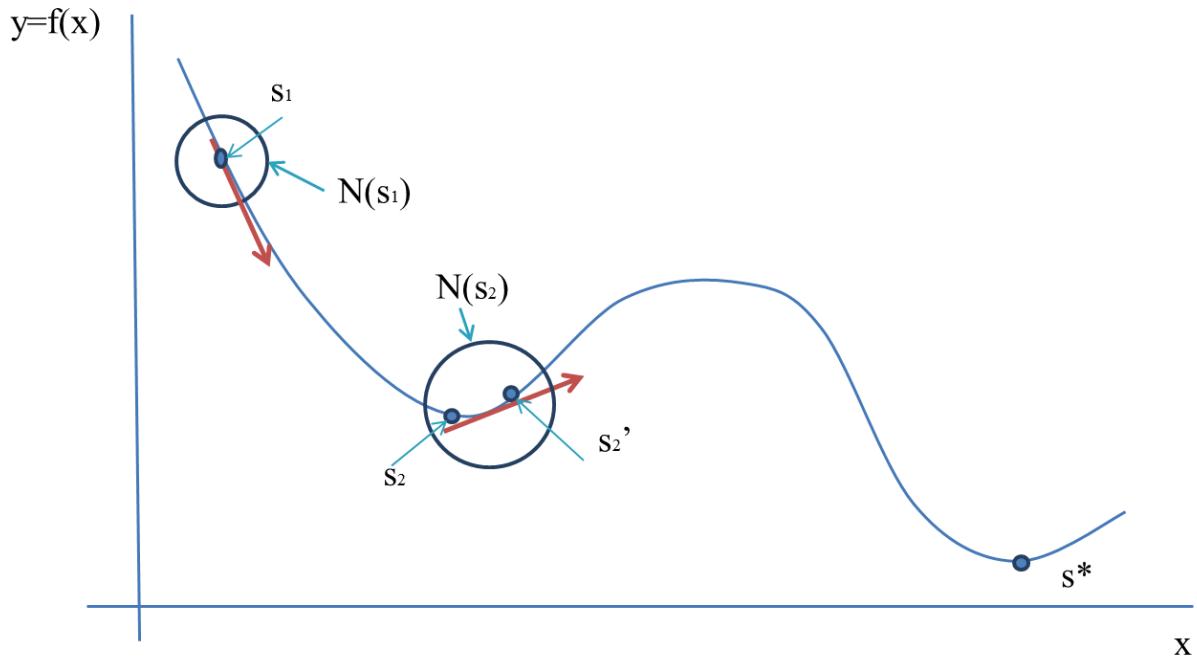
Set  $s^* = \hat{s}$

**If** termination criteria are satisfied, **do**:

Terminate

**Else:**

SEARCH( $s$ )



**Figure 5-1**

The descent method is one of the most straightforward approaches in that it only permits choosing neighbor solutions that improve the current objective value and stops when no improving solutions can be found. Figure 5-1 illustrates a descent method searching for a solution that minimizes  $f(x)$ . Note that in the figure, solution with subscript represents the solution in the specific iteration, i.e.  $s_1$  is the solution in iteration 1; and superscript, which will be used in Section 5.1, represents the index of the neighbor solution in a neighborhood, i.e.  $s^1$  is the first solution explored in a certain neighborhood. Suppose the algorithm starts from solution  $s_1$ , it selects the best solution out of neighborhood  $N(s_1)$  and descends along the function curve. The process terminates once it reaches local optimum  $s_2$  where no neighbor solution value is better, as a result, researchers have observed that such methods often do not find a globally optimal solution. In order to avoid getting stuck at a local optimum, tabu search (TS) is designed to accept non-improving solutions and to keep track of the information related to the best

solution as well as visited solutions during the search process ([23]). For example, Figure 5-1 shows that the search takes a worse solution  $s'_2$  as the next solution, assigns a tabu status to  $s_2$  and forbids choosing it again in a certain period of time. By doing so, TS is able to escape the trap of local optimum and to examine a more diverse solution space. In fact, TS has obtained optimal and near-optimal solutions to a wide range of problems ([17]). An alternative method would add a penalty cost to a solution that is similar to the tabu solution and choose the one that minimizes the penalized cost.

The essential compounds of a tabu search procedure include the neighborhood construction and solution selection. This thesis develops a TS procedure taking advantage of a new neighborhood, which is presented in the first section, and the detailed search mechanism is introduced in the following sections.

## 5.1 Route-based Neighborhood

It is obvious that for the integrated model IM which is

$$\min \sum_{\tau \in \theta} f_{\tau} z_{\tau}$$

subject to

$$\sum_{j:(i,j) \in \mathcal{A}} x_{ij}^k - \sum_{j:(i,j) \in \mathcal{A}} x_{ji}^k = \delta_i^k, \quad \forall i \in \mathcal{N}, \forall k \in \mathcal{K} \quad (1)$$

$$\sum_{k \in \mathcal{K}} d^k x_{(i,j),\tau}^k \leq u_{\tau} z_{\tau}, \quad \forall (i,j) \in \tau, \forall \tau \in \theta \quad (6)$$

$$\sum_{\tau \in \theta} x_{(i,j),\tau}^k = x_{ij}^k, \quad \forall (i,j) \in \mathcal{A} \quad (7)$$

$$z_\tau \in Z_+, \quad \forall \tau \in \theta$$

$$0 \leq x_{ij}^k \leq 1, \quad \forall k \in \mathcal{K}, \forall (i, j) \in \mathcal{A}$$

the objective function value is driven by the assignment of route design variables  $z$ . The neighborhood  $N(s)$  constructed here for a given solution  $s = (\bar{x}, \bar{z})$  corresponds to closing a number of open routes in the current solution  $s$  and reassigning flow on those routes to existing or new routes. Specifically, for a given solution  $s = (\bar{x}, \bar{z})$ ,  $\tilde{\Gamma}(s) = \{\tau \in \theta: \bar{z}_\tau > 0\}$  is defined as the set of open routes  $\tau$ . Among  $\tilde{\Gamma}(s)$ , every  $\tau$  (or multiple  $\tau$ ) may be closed, leading to various flow modification/deviation and changes in total cost and potentially a different solution  $s'$ . To build such a neighbor solution  $s'$ , IM is solved, with additional constraints that reflect the closing of certain routes in  $s$ .

For example, with current solution  $s$ , to obtain its neighbor solution  $s^1$  where one copy of  $\tau^1$  is closed, the restricted model  $\text{IM}(\tau^1)$  is shown below. Define the set of the deviated commodities on  $\tau^1$  is  $\mathcal{K}^1$ ,  $\text{IM}(\tau^1)$  ensures that  $z_{\tau^1} \leq \bar{z}_{\tau^1} - 1$  (constraint (14)), which is equivalent to restrict flow amount of  $\mathcal{K}^1$  on  $\tau^1$  to  $(\bar{z}_{\tau^1} - 1) * u_\tau$  (constraint (15)). The flow amount of commodities other than  $\mathcal{K}^1$  of the remaining open routes are fixed to their current value (constraint (16)).

$$\min \sum_{\tau \in \theta} f_\tau z_\tau$$

subject to

$$\sum_{j: (i,j) \in \mathcal{A}} x_{ij}^k - \sum_{j: (i,j) \in \mathcal{A}} x_{ji}^k = \delta_i^k, \quad \forall i \in \mathcal{N}, \forall k \in \mathcal{K} \quad (11)$$

$$\sum_{k \in \mathcal{K}} d^k x_{(i,j,\tau)}^k \leq u_\tau z_\tau, \quad \forall (i,j) \in \tau, \forall \tau \in \theta \quad (12)$$

$$\sum_{\tau \in \theta} x_{(i,j,\tau)}^k = x_{ij}^k, \quad \forall (i,j) \in \mathcal{A} \quad (13)$$

$$z_{\tau^1} \leq \bar{z}_{\tau^1} - 1 \quad (14)$$

$$\sum_{k \in \mathcal{K}^1} x_{(i,j,\tau^1)}^k \leq (\bar{z}_{\tau^1} - 1) * u_{\tau^1}, \quad \forall (i,j) \in \tau^1 \quad (15)$$

$$x_{(i,j,\tau)}^k = \bar{x}_{(i,j,\tau)}^k, \quad \forall (i,j) \in \tau, \forall \tau \in \tilde{\Gamma}(s) \setminus \tau^1, \forall k \in \mathcal{K} \setminus \mathcal{K}^1 \quad (16)$$

$$z_\tau \in Z_+, \quad \forall \tau \in \theta$$

$$0 \leq x_{ij}^k \leq 1, \quad \forall k \in \mathcal{K}, \forall (i,j) \in \mathcal{A}$$

In this algorithm, we also consider closing two routes that share nodes. They are referred as a pair, shown as  $\tau^1$  and  $\tau^2$  in Figure 5-2, because they are more likely to be combined and replaced by a third longer route ( $\tau^3$ ) which has less travel cost than the two routes added together. It is also true that this third route must not exceed the travel time regulation as well.

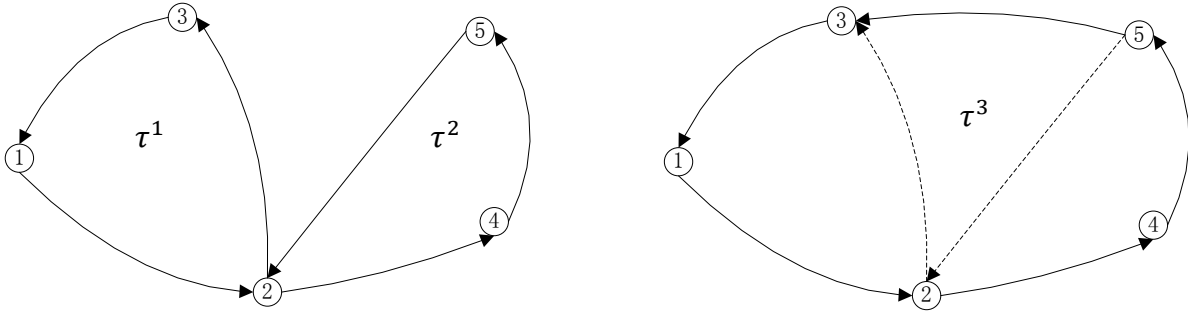


Figure 5-2

The restricted model becomes  $\text{IM}(\tau^1, \tau^2)$ . Formally, in an iteration with current solution  $s$  and best solution  $s^*$ , the search for neighbor solutions is written as follows.

**For**  $i \in \mathcal{N}$ , **do**



- a) Randomly choose  $\tau^1, \tau^2 \in \tilde{\Gamma}(s)$  that share the node  $i$
- b) Solve  $\text{IM}(\tau^1, \tau^2)$ , set solution as  $s^{1,2}$
- c) If  $f(s^{1,2}) < f(s^*)$ , stop and go to next iteration
- d)  $N(s) = N(s) \cup s^{1,2}$

**If**  $N(s) = \emptyset$ , **for**  $\tau \in \tilde{\Gamma}(s)$ , **do**

- a) Solve  $\text{IM}(\tau)$ , set solution as  $s'$
- b) If  $f(s') < f(s^*)$ , stop and go to next iteration
- c)  $N(s) = N(s) \cup s'$

## 5.2 Tabu Search

In our application, it is unnecessary to store a visited solution  $s = (\bar{x}, \bar{z})$  in its exact form in the tabu list mainly for two reasons. First, for instances we tested, the model is consisting of tens of thousands of variables, while only the variables of positive value are needed in the solution.

Keeping several solutions with a large number of values at a time consumes a great amount of memory and reduces computing speed. Second, the assignment of route design variables  $z$  is more important than flow variables  $x$  in terms of solution quality, since they determine the objective value. Also for a certain assignment  $\bar{z}$ , there may exist several different flow

assignments  $\bar{x}$  and they are considered as non-relevant information in terms of solution quality.

For example, if one treats  $\bar{s} = (\bar{x}, \bar{z})$  and  $\bar{s}^1 = (\bar{x}^1, \bar{z})$  as distinct elements, it becomes possible to select  $s^1$  from  $N(s)$  resulting no changes on solution cost because the  $\bar{z}$  vector remains same.

Therefore it is preferred to keep certain attributes related to a solution in the memory and the attributes should distinguish solutions without taking up much storage. For a solution  $s$ , an

attribute set  $B(s) = \{\sum_{\tau:(i,j) \in \tau} \bar{z}_{\tau} \text{ in } s: \forall (i,j) \in A\}$  is collected, and each attribute can be represented as  $(a, l)$ : on arc  $a$ , there are  $l$  number of open tours. As a result, the process labels attributes, instead of solutions, as tabu. To illustrate, Figure 5-3 shows a solution that has two routes open,  $1 \rightarrow 2 \rightarrow 3 \rightarrow 1$  and  $2 \rightarrow 3 \rightarrow 2$ , and its attribute set can be written as  $B(s) = \{1:(3,1), 1:(1,2), 2:(2,3), 1:(3,2)\}$ .

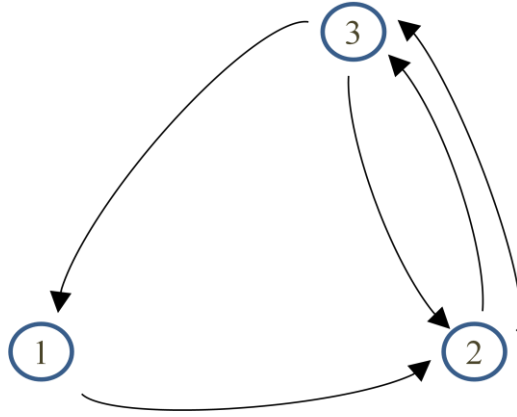


Figure 5-3

From  $N(s)$ , instead of choosing next solution by simply comparing objective function value, a candidate solution set  $M(s)$  is built using some more comprehensive criteria. For these candidate solutions, their solution cost are adjusted according to how many of their attributes have appeared in recently visited solutions. The adjustment equation is presented in the pseudocode of the whole process at the last part of this section. The next solution  $s'$  is then selected from this candidate set, based on the adjusted cost. This technique will help the process to identify a different structured solution so that it is more likely to explore a wider solution region. Figure 5-4 shows the relationship between the neighborhood  $N(s)$ , candidate set  $M(s)$  and the selected solution  $s'$ .

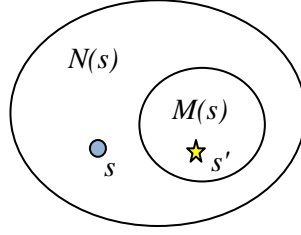


Figure 5-4

To formally describe the construction of  $M(s)$  and selection of  $s'$ , several notations in Table 5-1 regarding the solution attributes need to be introduced first. Each attribute  $(a, l)$  has three marks:  $\alpha_{al}$ , number of times that attribute  $(a, l)$  has been added to the current solution;  $\beta_{al}$ , last iteration for which  $(a, l)$  is declared tabu;  $\gamma_{al}$ , aspiration level of  $(a, l)$ .  $\beta_{al}$  is introduced to examine a solution's tabu status. A solution from  $N(s)$  will be added to  $M(s)$  when it has at least one non-tabu attribute.  $\gamma_{al}$  shows the best solution value whose  $B(s)$  contains  $(a, l)$ . A solution will also be added to  $M(s)$  when its value exceeds the aspiration level of at least one of its attributes. The use of  $\alpha_{al}$  is to compare a candidate solution's similarity with recently visited solutions. In other words, the more common attributes a solution shares with previous visited solutions, the less likely it is chosen from  $M(s)$ . Specifically we quantify the penalty based on the most frequently added attributes to the solution. The penalty is proportional to  $f(\hat{s})$ ,  $\alpha_{al}$ , and  $\sqrt{|\theta|\rho}$ , which is adapted from Cordeau et al [6]. The motivation of using  $\sqrt{|\theta|\rho}$  is as follows: since there are  $|\theta|$  number of routes, and each route has  $\rho$  arcs on average, the total number of possible attributes will be  $O(|\theta|\rho)$ , so the frequency of adding one given attribute to the current solution is inversely related to problem size, and using square root can adequately compensate for problem size.

Notations:

$f(s)$	solution value of $s$
$i$	Iteration counter
$\lambda$	Total number of iterations
$\rho$	Average number of arcs in a route
$\delta$	Tabu list length
$\sigma$	Number of non-improving iterations after which the route set get expanded
$\alpha_{al}$	Number of times that attribute $(a, l)$ has been added to the current solution
$\beta_{al}$	Last iteration for which $(a, l)$ is declared tabu
$\gamma_{al}$	Aspiration level of $(a, l)$

**Table 5-1**

With a route set expansion procedure (step g) as explained in next section, the overall tabu search process is as follows.

Initialization:

$$s = s^* = IPsol, i = 0$$

**For** every  $(a, l)$ , **do**     *#Define the attributes and their marks*

- a) Set  $\alpha_{al} = 0, \beta_{al} = 0$
- b) If  $(a, l) \in B(s)$ , set  $\gamma_{al} = f(s)$

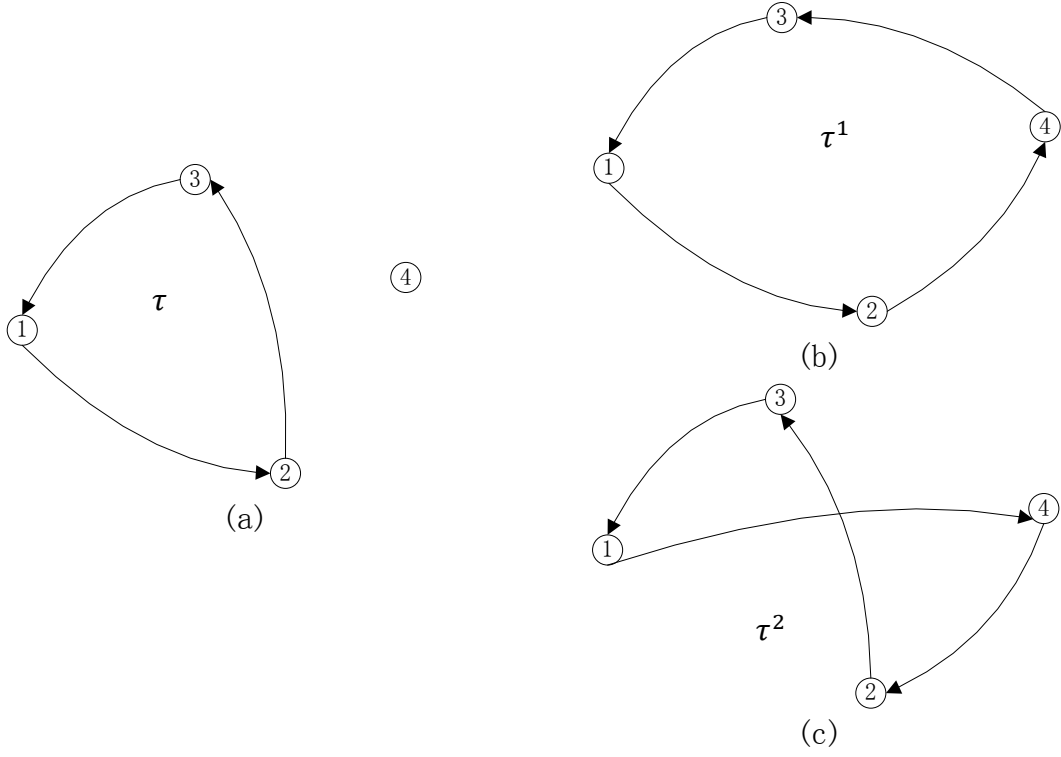
**For**  $i = 1 \dots \lambda$  **do**

- a) Set  $M(s) = \emptyset$      *#Initialize the candidate set*

- b) For  $\hat{s} \in N(s)$ , if there exists  $(a, l) \in B(\hat{s}) \setminus B(s)$  such that  $\beta_{al} < i$  or  $f(\hat{s}) < \gamma_{al}$ ,  
 set  $M(s) = M(s) \cup \{\hat{s}\}$       *#Add neighbor solution to candidate set if it has  
 non-tabu attribute or the solution value is smaller than at least one of its  
 attributes' aspiration level*
- c) For  $\hat{s} \in M(s)$ , if  $f(\hat{s}) \geq f(s)$ , set  $g(\hat{s}) = f(\hat{s}) * \left(1 + \sqrt{|\theta|\rho} * \sum_{al \in B(\hat{s})/B(s)} \frac{\alpha_{al}}{i}\right)$ ,  
 else set  $g(\hat{s}) = f(\hat{s})$       *#Penalize the candidates value based on its  
 similarity with recently visited solution if its value is no better than the current  
 solution;*
- d) Find solution  $s' \in M(s)$  that minimizes  $g(\hat{s})$       *#Choose the candidate with  
 smallest penalized cost as the next solution*
- e) For  $(a, l) \in B(s')$ ,  $\alpha_{al} = \alpha_{al} + 1, \beta_{al} = i + \delta$       *#Update attribute marks*
- f) If  $f(s') < f(s^*)$ , set  $s^* = s', i^* = i$ , for  $(a, l) \in B(s')$ ,  $\gamma_{al} = \min \{\gamma_{al}, f(s')\}$ .  
*#Update attributes aspiration level*
- g) If  $i - i^* > \sigma$ , expand  $\theta$ .      *#Expand the route set*

### 5.3 Route Set Expansion

A route set  $\theta$  expansion technique is included in the tabu search to diversify the solution search. The idea is to bring in more new routes to the problem after there has been no improvement on best solution for a period of time. Taking every open route from the current solution, the algorithm examines if there exists a single node that can be inserted into the route and create new routes potentially. Take Figure 5-5 for example, suppose  $\tau: 1 \rightarrow 2 \rightarrow 3 \rightarrow 1$  in (a) is open in the current solution, if there are arc (2,4) and (4,3) in  $\mathcal{A}$ , then  $\tau^1: 1 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 1$  in (b) is created, or if arc (2,4) and (1,4) are in  $\mathcal{A}$ ,  $\tau^2: 1 \rightarrow 4 \rightarrow 2 \rightarrow 3 \rightarrow 1$  in (c) is created.



**Figure 5-5**

Newly created routes are again brought into  $\theta$  and the neighborhood  $N(s)$  in the next iteration will be constructed using the expanded  $\theta$ .

## 6 RESULTS AND ANALYSIS

Because there is no existing algorithm for IM, to analyze the performance of the proposed approach, comparisons are made against the MIP solver, Gurobi Optimizer. We wish to use small enough instances that Gurobi could produce optimal or near-optimal solutions. With those solutions we can evaluate the quality of the solutions produced by the heuristic proposed. Two sets of instances are generated, each has 10. Instances in one set have 20 nodes, 100 arcs and 15 commodities (20N,100A,15C) and the second set has instances with 40 nodes, 200 arcs and 15 commodities (40N,200A,15C). In these instances, each arc has random length ranging from 150 to 250 units.

The experiments of each instance are done on 3 groups with different route length limit, 1000, 1200 and 1400 units, which therefore defined the algorithm parameter  $\rho$  - average number of arcs in a route as 5,6,7 respectively for the groups. Each IP solved in the course of executing the SS and tabu procedures is solved to 1% optimality or 10 seconds time limit. Tabu list length  $\delta$  is 7 for all tests, and the route set get expanded after  $\sigma = 15$  non-improving iterations. MIP solving is set to terminate after either running for 960 minutes or reaching 1% optimality gap and the heuristic terminates after 960 minutes or 500 iterations. Both programs using the heuristic and Gurobi solver are written in Python. The experiments reported are performed on a computing cluster system. Each run of an instance was assigned to one of the nodes that have 32-64 cores with AMD Opteron 2.2 GHz or AMD Interlagos 2.6 "bulldozer" processors and 128-256 GB of memory.

The average result of Gurobi versus the heuristic is presented in Table 6-1. GUROBI Solution Time to Best denotes the CPU time (in minute) used to reach the best solution (5 hours

on average) and Opt. Gap reported by Gurobi (25.63% on average). ROUTE-TABU columns contain the heuristic results. Gap w/GUROBI denotes in percentage the difference between best solutions obtained from tabu search and Gurobi best solutions, namely  $GAP =$

$\frac{TABU - GUROBI}{GUROBI} * 100\%$  and Gap w/Bound represents the difference from Gurobi best bound. With

an average 5.36% improvement compared to Gurobi and 18.84% away from best bound, the proposed search is able to find better solutions than the MIP solver after running for the same length of time. In addition, the proposed heuristic provided a 2.29% improvement with regards to the best solution reported by Gurobi when Gurobi reached its best. Results by instance are presented in Appendix A and Appendix B.

Instance Set	Route Length Limit	GUROBI Solution		ROUTE-TABU Solution			
		GUROBI Solution Time to Best (min)	Opt. Gap	Gap w/ GUROBI	Gap w/Bound	Time (min)	Solution gap when Gurobi found best
20N,100A,15C	1000	232	17.49%	-5.32%	11.17%	485	-1.46%
	1200	301	17.84%	-4.96%	11.99%	435	-3.43%
	1400	354	23.16%	-4.36%	17.66%	574	0.62%
	<b>AVERAGE</b>	<b>296</b>	<b>19.50%</b>	<b>-4.88%</b>	<b>13.60%</b>	<b>498</b>	<b>-1.42%</b>
40N,200A,15C	1000	346	31.35%	-4.24%	25.80%	457	-2.35%
	1200	247	32.46%	-6.05%	24.40%	467	-3.45%
	1400	288	31.49%	-7.22%	22.02%	524	-3.65%
	<b>AVERAGE</b>	<b>294</b>	<b>31.77%</b>	<b>-5.84%</b>	<b>24.08%</b>	<b>483</b>	<b>-3.15%</b>
<b>AVERAGE</b>		<b>295</b>	<b>25.63%</b>	<b>-5.36%</b>	<b>18.84%</b>	<b>490</b>	<b>-2.29%</b>

Table 6-1

Table 6-2 presents SS INITIAL versus TABU results. It is seen that SS yields good quality initial solutions, approximately 12.38% away from Gurobi, within 20~30 minutes. And the heuristic improves the initial solutions by 15.33%.



Instance Set	Route Length Limit	SS INITIAL GAP w/ GUROBI	SS TIME (min)	ROUTE-TABU Solution GAP w/ INITIAL
20N,100A,15C	1,000	13.53%	11	-16.31%
	1,200	16.00%	12	-17.81%
	1,400	18.26%	30	-18.26%
	<b>AVERAGE</b>	<b>15.93%</b>	<b>18</b>	<b>-17.46%</b>
40N,200A,15C	1,000	10.94%	5	-13.44%
	1,200	6.86%	9	-11.95%
	1,400	8.71%	35	-14.20%
	<b>AVERAGE</b>	<b>8.84%</b>	<b>16</b>	<b>-13.20%</b>
<b>AVERAGE</b>		<b>12.38%</b>	<b>17</b>	<b>15.33%</b>

Table 6-2

Table 6-3 shows that each iteration consumed on average 159 seconds on small instances and 185 seconds on large ones, and about 40%~60% of the total time is spent on building neighborhood solutions  $N(s)$ . And no more than 10% of total iterations found improving solutions, which suggests that there are possibilities of enhancing the search technique so that it can be more efficient and collecting more promising solutions.

Instance Set	Route Length Limit	Average Time of Each Iteration	% Improving iters	% Time Defining $N(s)$
20N,100A,15C	1000	130	4.20%	66.87%
	1200	172	4.68%	58.45%
	1400	187	6.34%	52.32%
40N,200A,15C	1000	140	5.11%	56.02%
	1200	188	5.48%	43.55%
	1400	226	8.73%	44.33%

Table 6-3

The quality of the routes generated after SS and after ROUTE-TABU is examined by running MIP on the two sets respectively. Time limit is set to 10,000 seconds and optimality gap 1%. Table 6-4 shows that the solutions are superior to the ones of the complete route set. And

MIP on SS routes yields 13% better than the initial solution provided by SS, though still with about 20% optimality gap. Time to reach best solution from ROUTE-TABU routes is longer because the problem becomes larger due to route set expansion. These results indicate that two procedures are able to produce very useful routes.

Instance Set	Route Length Limit	MIP on SS Routes Gap w/ GUROBI	TIME (min)	MIP on SS Routes Gap w/ INITIAL	MIP on ROUTE-TABU Routes Gap w/ GUROBI	TIME (min)
20N,100A,15C	1000	-2.64%	31	-13.76%	-3.77%	66
	1200	-2.84%	28	-15.78%	-4.96%	81
	1400	-2.04%	38	-15.97%	-5.61%	66
40N,200A,15C	1000	-2.21%	60	-11.45%	-4.09%	81
	1200	-4.98%	78	-10.64%	-6.99%	55
	1400	-4.95%	75	-11.74%	-6.12%	111

**Table 6-4**

Sensitivity analysis is performed on parameter  $\sigma$  – the number of non-improving iterations after which the route set is expanded.  $\sigma$  is set to 5 and 10 respectively and the results in Table 6-5 suggest that creating routes more frequently does not improve the solution for all the instances. The reason might be a larger route set will require the restricted IP more time to explore the solution space, therefore the solution found in 10 seconds time limit will not be as good as the one from a smaller solution space.

Instance Set	Route Length Limit	$\sigma = 5$ GAP w/ $\sigma = 15$	$\sigma = 10$ GAP w/ $\sigma = 15$
20N,100A,15C	1,000	1.27%	0.25%
	1,200	-0.90%	-0.28%
	1,400	0.77%	1.06%
40N,200A,15C	1,000	0.75%	-0.85%
	1,200	0.24%	-1.25%
	1,400	0.29%	0.80%

**Table 6-5**

To show the tabu search performance over time, solution value and time (in minute) of one instance was collected and presented in Figure 6-1. It is seen that within the beginning of time, the best solution is constantly improved, and the speed of finding better solution slows down along the process.

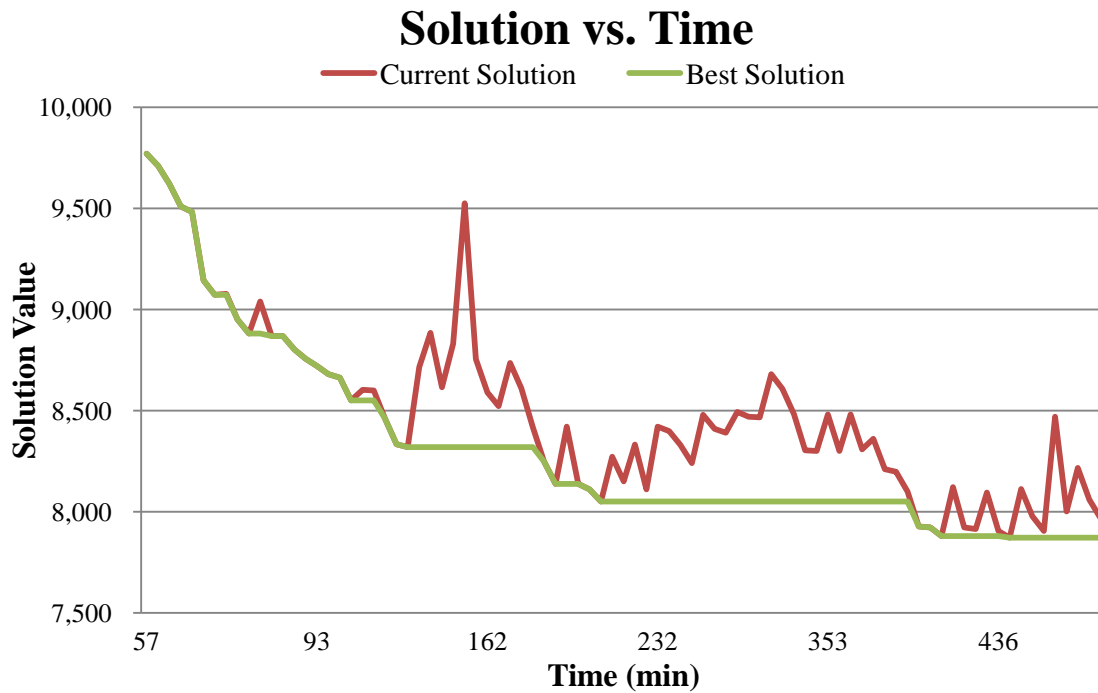


Figure 6-1

## 7 CONCLUSIONS AND FUTURE WORK

This thesis presented a heuristic algorithm (ROUTE-TABU) for solving the integrated model of consolidation network design problem. The problem formulation and solution technique can be adapted to various sets of driving regulations and rules, as well as route capacity and shipment demand, with the optimization goal being minimizing the total cost incurred by installing the routes. The initial solution for the heuristic is provided by a slope scaling which alternates between solving linear relaxations of the original problem and generating routes. Then the algorithm uses a tabu search scheme to improve upon the best solution found during the slope scaling procedure. To find a neighbor of the current solution, a route in the current solution is closed and an IP is solved to reroute its flow. The procedure selects a neighbor from the candidates as next solution, based on both its objective function value and its similarity to previously visited solutions..

The proposed method was tested and compared against a commercial MIP solver because there is no existing algorithm for the problem. Results showed that the tabu search provides solutions that are on average 5% better than the solver within in the same period of time, and it finds better solution (2%) at the time when the solver reaches its best solution. It is concluded that on the instances tested, ROUTE-TABU outperforms the MIP solver in terms of solution quality and time efficiency.

However the results also indicated that each iteration of the tabu search took 3~4 minutes, and the percentage of iterations that found improved solution is small, no more than 10% of the total number of iterations. These performance indicators showed that there is still opportunity to refine the technique to find neighbor solutions, collect attributes and calculate the penalized cost.

In addition to improving efficiency by designing a better scheme, it is also possible to make improvements on implementation of the code.

Also from the limited size and number of instances, it is hard to prove that ROUTE-TABU has better performance on larger or different-structured instances. More future work can be done using different instances. In addition, the idea of considering HOS regulations can be extended to other similar resources limit, such as different country's driving rules or the company's own business laws.

## REFERENCES

- [1] American Transportation Research Institute, "An analysis of the operational costs of trucking," 2012.
- [2] American Trucking Association, "Trucking and the economy," 2008.
- [3] J. Andersen, M. Christiansen, T. G. Crainic, and R. Gronhaug, "Branch and price for service network design with asset management constraints," *TRANSPORTATION SCIENCE*, vol. 45, pp. 33-49, February 1, 2011 2011.
- [4] J. Andersen, T. G. Crainic, and M. Christiansen, "Service network design with asset management: Formulations and comparative analyses," *Transportation Research Part C: Emerging Technologies*, vol. 17, pp. 197-207, 4// 2009.
- [5] Bureau of Transportation Statistics, "Commodity flow survey United States 2007," 2010.
- [6] J. F. Cordeau, M. Gendreau, and G. Laporte, "A tabu search heuristic for periodic and multi-depot vehicle routing problems," *Networks*, vol. 30, pp. 105-119, 1997.
- [7] T. H. Cormen and Books24x7 Inc. (2009). *Introduction to algorithms, third edition (3rd ed.)* [Text]. Available: <https://ezproxy.rit.edu/login?url=http://library.books24x7.com/library.asp?^B&bookid=49924>
- [8] T. G. Crainic, "Service network design in freight transportation," *European Journal of Operational Research*, vol. 122, pp. 272-288, 4/16/ 2000.
- [9] T. G. Crainic, "Long-haul freight transportation," in *Handbook of transportation science*, R. W. Hall and ebrary Inc., Eds., 2nd ed Boston: Kluwer Academic Publishers, 2003, pp. 451-516.
- [10] T. G. Crainic, M. Gendreau, and J. M. Farvolden, "A Simplex-based tabu search method for capacitated network design," *INFORMS J. on Computing*, vol. 12, pp. 223-236, 2000.
- [11] T. G. Crainic, B. Gendron, Genevi, and v. Hernu, "A Slope Scaling/Lagrangian Perturbation Heuristic with Long-Term Memory for Multicommodity Capacitated Fixed-Charge Network Design," *Journal of Heuristics*, vol. 10, pp. 525-545, 2004.
- [12] A. Erera, M. Hewitt, M. Savelsbergh, and Y. Zhang, "Improved load plan design through integer programming based local search," 2009.
- [13] D. Feillet, P. Dejax, and M. Gendreau, "The profitable arc tour problem: solution with a branch-and-price algorithm," *TRANSPORTATION SCIENCE*, vol. 39, pp. 539-552, November 1, 2005 2005.

- [14] B. Gendron, T. Crainic, and A. Frangioni, *Multicommodity capacitated network design*: Centre for Research on Transportation= Centre de recherche sur les transports (CRT), 1998.
- [15] I. Ghamlouche, T. G. Crainic, and M. Gendreau, "Cycle-based neighborhoods for fixed-charge capacitated multicommodity network design," *Operations Research*, vol. 51, pp. 655-667, 2003.
- [16] Global Insights Inc., "The U.S. truck driver shortage: analysis and forecasts," ed, 2005.
- [17] F. Glover, "Tabu search : a tutorial," *Interfaces*, vol. 20, pp. 74-94, 1990 1990.
- [18] A. Goel, "Vehicle scheduling and routing with drivers' working hours," *Transportation Science*, vol. 43, pp. 17-26, 2009.
- [19] A. Goel and T. Vidal, "Hours of service regulations in road freight transport: an optimization-based international assessment," 2012.
- [20] M. Haouari, "Arc Routing: Theory, Solutions and Applications (Book)," *IIE Transactions*, vol. 34, pp. 331-332, 2002.
- [21] A. Hertz, "Recent trends in arc routing," *Graph theory, combinatorics and algorithms*, pp. 215-236, 2005.
- [22] A. Hertz, G. Laporte, and M. Mittaz, "A Tabu Search Heuristic for the Capacitated Arc Routing Problem," *Operations Research*, vol. 48, pp. 129-135, 2000.
- [23] A. Hertz, E. Taillard, and D. de Werra, "A tutorial on tabu search," in *Giornate di Lavoro*, 1995, pp. 13-24.
- [24] M. Hewitt, "Scheduling drivers and routing shipments in tandem," in *Industrial Engineering Research Conference*, 2010.
- [25] M. Hewitt, G. L. Nemhauser, and M. W. P. Savelsbergh, "Combining exact and heuristic approaches for the capacitated fixed-charge network flow problem," *INFORMS Journal on Computing*, September 21, 2009 2009.
- [26] A. I. Jarrah, E. Johnson, and L. C. Neubert, "Large-scale less-than-truckload service network design," *Operations Research*, vol. 57, pp. 609-625, 2009.
- [27] D. Kim, X. Pan, and P. M. Pardalos, "An enhanced dynamic slope scaling procedure with tabu scheme for fixed charge network flow problems," *Comput. Econ.*, vol. 27, pp. 273-293, 2006.
- [28] T. L. Magnanti and R. T. Wong, "Network design and transportation planning: models

- and algorithms," vol. 18, ed. Transportation Science, 1986, pp. 1-55.
- [29] M. B. Pedersen, T. G. Crainic, and O. B. G. Madsen, "Models and tabu search metaheuristics for service network design with asset-balance requirements," *TRANSPORTATION SCIENCE*, vol. 43, pp. 158-177, May 1, 2009 2009.
  - [30] W. B. Powell, "A local improvement heuristic for the design of less-than-truckload motor carrier networks," *Transportation Science*, vol. 20, pp. 246-357, 1986.
  - [31] M.-E. Rancourt, J.-F. Cordeau, and G. Laporte, "Long-haul vehicle routing and scheduling with working hour rules," *Transportation Science*, June 22, 2012 2012.
  - [32] US Department of Transportation, "Hours of service of drivers," Lanham, United States, Lanham2011.



## APPENDICES – DETAILED RESULTS

### APPENDIX A

Instance Set	Route Length Limit	Instance	GUROBI		SS		ROUTE-TABU					
			TIME (min)	Opt Gap	GAP w/ Gurobi	TIME (min)	GAP w/ GUROBI	GAP w/ SS	TIME (min)	# Improving Iter./Total iter.	Gap when GUROBI found best	Avg. Time/Iter.
20N, 100A, 15C	1,000	1	27	18.30%	13.38%	4	-5.91%	17.02%	539	21/344	2.56%	141
		2	819	14.74%	13.88%	10	0.00%	12.19%	688	16/258	0.00%	191
		3	55	15.42%	8.17%	7	-7.33%	14.33%	254	11/410	-5.90%	107
		4	110	11.22%	30.88%	7	-0.54%	24.01%	522	22/438	6.57%	120
		5	169	22.87%	15.56%	45	-5.94%	18.60%	846	22/297	6.60%	150
		6	126	20.80%	1.50%	22	-5.56%	6.95%	278	9/350	-5.41%	161
		7	76	15.48%	17.32%	1	-6.10%	19.97%	682	18/499	-4.08%	107
		8	53	20.48%	2.09%	3	-11.15%	12.97%	103	10/500	-7.16%	80
		9	48	17.05%	15.04%	6	-2.57%	15.31%	792	13/464	0.38%	106
		10	841	18.56%	17.46%	3	-8.15%	21.80%	144	13/357	-8.15%	135
	<b>AVERAGE</b>		<b>232</b>	<b>17.49%</b>	<b>13.53%</b>	<b>11</b>	<b>-5.32%</b>	<b>16.31%</b>	<b>485</b>	<b>-</b>	<b>-1.46%</b>	<b>130</b>
	1,200	1	108	17.66%	7.17%	26	-2.65%	9.16%	102	9/247	-2.65%	177
		2	916	17.15%	30.01%	8	-1.79%	24.45%	350	16/215	-1.79%	216
		3	101	18.10%	6.91%	8	-10.51%	16.29%	189	11/342	-8.73%	130
		4	158	15.32%	20.88%	6	-3.41%	20.09%	322	11/288	1.11%	164
		5	399	17.55%	33.59%	24	3.34%	22.65%	690	24/209	3.79%	225
		6	13	22.85%	14.03%	9	-6.74%	18.22%	556	12/219	-1.13%	214
		7	189	14.98%	12.54%	6	-8.45%	18.65%	707	11/331	-7.27%	154
		8	103	18.10%	10.28%	4	-8.39%	16.93%	491	10/361	-7.30%	130
		9	731	16.02%	12.55%	15	-6.47%	16.90%	863	11/265	-5.78%	170
		10	293	20.62%	12.06%	15	-4.52%	14.79%	78	6/400	-4.52%	140
	<b>AVERAGE</b>		<b>301</b>	<b>17.84%</b>	<b>16.00%</b>	<b>12</b>	<b>-4.96%</b>	<b>17.81%</b>	<b>435</b>	<b>-</b>	<b>-3.43%</b>	<b>172</b>
	1,400	1	374	20.55%	10.82%	39	-2.97%	12.44%	511	14/217	-0.94%	173
		2	574	17.61%	10.67%	91	-2.63%	12.02%	536	15/171	-2.63%	232
		3	599	22.50%	3.78%	22	-10.86%	14.10%	355	15/312	-10.86%	139

		4	731	18.98%	49.54%	9	-1.48%	34.12%	401	20/321	-1.48%	150
		5	34	24.15%	19.13%	58	-4.35%	19.71%	736	15/184	19.13%	215
		6	202	19.55%	21.16%	17	-0.91%	18.21%	857	13/188	1.24%	236
		7	192	16.64%	9.69%	3	1.43%	7.53%	796	13/299	3.16%	168
		8	15	21.04%	12.64%	12	-9.43%	19.60%	718	18/293	8.09%	139
		9	546	52.14%	36.09%	27	-7.00%	31.66%	336	12/219	-7.00%	212
		10	278	18.43%	9.03%	23	-5.42%	13.25%	494	12/197	-2.54%	208
		AVERAGE		354	23.16%	18.26%	30	-4.36%	18.26%	574	-	0.62%

## APPENDIX B

Instance Set	Route Length Limit	Instance	GUROBI		SS		ROUTE-TABU					
			TIME (min)	Opt Gap	GAP w/ Gurobi	TIME (min)	GAP w/ GUROBI	GAP w/ SS	TIME (min)	# Improving Iter./Total iter.	Gap when GUROBI found best	Avg. Time/Iter.
40N, 200A, 15C	1000	11	12	31.46%	4.48%	7	-10.76%	14.58%	570	15/415	-3.23%	120
		12	400	36.86%	20.57%	8	-1.63%	18.41%	611	28/242	0.72%	199
		13	777	35.63%	10.19%	3	-1.92%	11.00%	198	18/362	-1.92%	157
		14	739	34.91%	24.68%	5	-4.01%	23.02%	812	28/275	-3.95%	178
		15	175	33.18%	10.53%	2	-4.17%	13.30%	864	16/500	-2.98%	99
		16	336	27.99%	2.95%	4	-1.62%	4.44%	481	7/364	-1.31%	123
		17	523	27.64%	6.09%	5	-4.90%	10.35%	65	8/331	-4.90%	128
		18	1	29.06%	8.00%	10	-7.10%	13.98%	212	17/271	-	163
		19	285	28.90%	14.74%	1	-0.26%	13.07%	167	15/427	-0.26%	127
		20	216	27.82%	7.14%	1	-6.01%	12.27%	591	14/409	-3.29%	110
	<b>AVERAGE</b>		<b>346</b>	<b>31.35%</b>	<b>10.94%</b>	<b>5</b>	<b>-4.24%</b>	<b>13.44%</b>	<b>457</b>	<b>-</b>	<b>-2.35%</b>	<b>140</b>
	1200	11	22	28.93%	2.87%	14	-7.88%	10.45%	492	13/247	-3.02%	176
		12	49	37.61%	7.48%	13	-4.99%	11.60%	686	14/190	-0.29%	228
		13	889	29.09%	15.84%	7	-1.33%	14.82%	711	18/223	-1.33%	214
		14	471	40.34%	10.58%	13	-5.47%	14.52%	507	24/235	-3.96%	194
		15	335	40.62%	2.66%	3	-7.44%	9.84%	16	6/346	-7.44%	146

		16	360	26.53%	-0.42%	7	-0.64%	0.22%	23	2/313	-0.75%	168
		17	96	29.13%	7.58%	5	-4.73%	11.44%	122	12/246	-2.99%	192
		18	52	28.30%	12.42%	18	-6.22%	16.58%	933	15/181	-0.34%	245
		19	124	32.81%	6.71%	5	-9.50%	15.19%	549	14/261	-4.67%	184
		20	76	31.24%	2.90%	4	-12.34%	14.82%	633	9/304	-9.69%	130
	<b>AVERAGE</b>		<b>247</b>	<b>32.46%</b>	<b>6.86%</b>	<b>9</b>	<b>-6.05%</b>	<b>11.95%</b>	<b>467</b>	<b>-</b>	<b>-3.45%</b>	<b>188</b>
	1400	11	418	27.78%	-5.10%	26	-10.88%	6.09%	129	11/180	-10.88%	186
		12	342	40.69%	9.76%	48	-5.68%	14.07%	958	17/165	-0.98%	225
		13	451	34.01%	20.77%	58	-5.20%	21.50%	756	30/137	-2.71%	321
		14	50	37.44%	25.01%	7	-5.14%	24.12%	293	21/244	-0.32%	187
		15	22	29.32%	7.21%	6	-5.38%	11.74%	141	7/253	-0.90%	159
		16	69	28.09%	-1.50%	20	-10.75%	9.39%	381	5/184	-1.50%	205
		17	4	30.94%	11.53%	72	-10.13%	19.42%	957	18/150	-	261
		18	296	25.38%	9.70%	45	-4.35%	12.80%	755	12/129	-1.14%	282
		19	895	28.71%	11.62%	9	-4.94%	14.84%	476	12/213	-4.94%	190
		20	336	32.52%	-1.86%	60	-9.72%	8.01%	399	9/143	-9.50%	244
	<b>AVERAGE</b>		<b>288</b>	<b>31.49%</b>	<b>8.71%</b>	<b>35</b>	<b>-7.22%</b>	<b>14.20%</b>	<b>524</b>	<b>-</b>	<b>-3.65%</b>	<b>226</b>