12-2011

Empirical Analysis of the Relay Network Design Problem

Sergio Maldonado Soria Galvarro
University of Arkansas, Fayetteville

Empirical Study of the Relay Network Design Problem

Sergio Maldonado

Undergraduate Honors Student in Industrial Engineering

University of Arkansas

Department of Industrial Engineering

December 12, 2011

Advisor:

Sarah Root, Ph.D.

Reader:

Chase Rainwater, Ph.D.

1. Introduction and Problem Description

Trucking plays a highly important role in the transportation industry accounting for the movement of approximately $8.3 trillion worth of freight in a year (U.S. Bureau of Transportation Statistics (2009)). Despite its importance in our economy, there are still significant problems within the trucking industry. One of the major problems for trucking companies is a significant driver shortage, currently estimated to be 125,000 drivers in the United States (Morris Frank, 2011). This is exacerbated by high driver turnover rates. Given the importance that trucking plays in the U.S. commercial freight transportation industry, research in this critical industry is necessary.

There are two primary segments of the trucking industry: less-than-truckload (LTL) shipping and truckload (TL) shipping. LTL operations move small shipments (typically less than 10,000 lbs.) between a variety of origin-destination (O-D) pairs. These small shipments travel through a network of locations where they are sorted and consolidated to create more highly-utilized loads for transport and distribution. Alternatively, TL operations use a system which consists of direct service from origin to destination also known as Point-to-Point dispatching (PtP). After TL drivers deliver a load, they must travel to the origin of their next load. Often, this requires them to move a considerable distance without a load attached to the truck; such movements are called dead-head trips. Creating a sequence of loaded and dead-head trips that return TL drivers to their homes often requires drivers considerable amounts of time on the road. It is not uncommon for TL drivers to be on the road for two to three weeks at a time. This negatively impacts drivers' quality of life, and leads to high driver turnover. In 2007, the line-haul driver turnover rate for large TL carriers was 112% (American Trucking Associations, 2008). This is a clear reflection of the long times drivers spend on the road, and consequently,

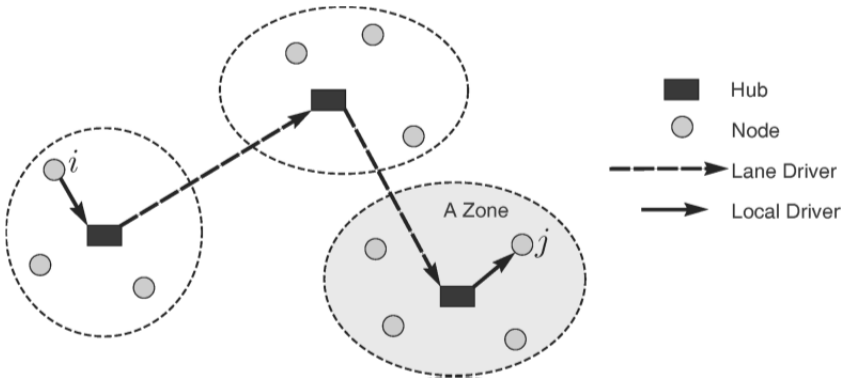
away from their homes. Due to their poor quality of life, drivers decide to change jobs or quit, requiring TL carriers to consistently invest a significant amount of money into the training and recruitment of new drivers.

In contrast, turnover for LTL carriers is around 15%. LTL drivers typically move shipments between hubs in the consolidation network. These trips are shorter and occur between a limited number of facilities. As a result, drivers are able to return home more frequently, positively impacting driver retention particularly as compared to TL carriers.

This significant difference in driver turnover between TL and LTL carriers is the motivation for this research and related research that considers alternative methodologies for Point-to-Point dispatching. Our research considers the movement of TL loads through a network containing different relay points (RPs). At RPs, drivers and trailers would be exchanged much like they are in LTL operations (Campbell 2005). Unlike LTL operations, however, freight on the loads would not be sorted or consolidated. RPs would serve solely as points where drivers would be exchanged. In order to transport loads from origin to destination, each truckload would visit one or more different relay points (RPs) along the network.

Relay networks produce noticeable advantages due to the reduction in backhauls, better truck utilization, and potential reduction of total delivery time (Taylor et al., 2001). In addition, relay networks produce an increase in tour regularity and reduce of driver tour length.

As shown in Figure 1, when implementing relay networks, origins and destinations are assigned to *zones* served by a single RP (Taylor et al., 2001). Two types of drivers are used to transport loads through a relay network. Local drivers are responsible for moving loads within a specific zone without leaving the boundary of this zone (i.e., origin node to RP, or RP to destination node). Distances traveled by local drivers typically do not exceed 150 miles. Lane drivers are responsible for inter-zone movement (i.e., from one RP to another). These drivers travel longer distances than local drivers. However, the distances between relay points allow drivers travel between relay points without exceeding federal hours of service regulations.



Source: Üster and Maheshwari, 2007.

Figure1. Multi-zone dispatching example.

To use a relay network, TL carriers must determine where to place RPs, and how to route the loads through these RPs. Routes taken by loads in the network must not violate operational constraints such as limitations on circuity and the maximum number of relay points that a load can visit. These are challenging to handle from a mathematical perspective, yet critical from an implementation perspective. The objective of TL carriers is to minimize the overall cost (i.e., fixed costs + variable transportation costs) of operating the network. The increase in cost from

installing RPs must be carefully balanced against the savings from increased driver retention to produce overall savings for carriers.

2.1. Relay Network Design Model

To solve the relay network design problem (RNDP), we use a mathematical modeling approach (Vergara, H. (2010)). This approach uses a composite variable model to capture the RNDP. In this model, the variables represent a feasible routing for a truckload. Note that these variables are also referred to as composites.

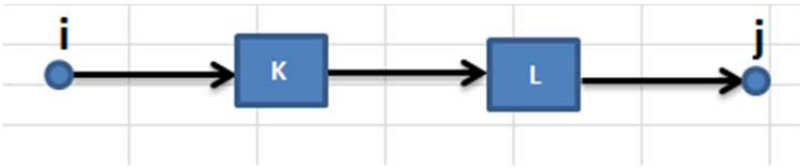


Figure 2. Feasible route example.

Figure 2 shows an example of a composite variable for a load moving from node i to node j. This load stops at relay points K and L where drivers are exchanged. In order to be feasible, we must ensure that the local and lane distances are not exceeded, and that the number of relay points visited does not exceed the permissible number. Given these constraints that govern which routes are feasible, we generate all possible feasible routes to be used as composite variables in our model. This will be discussed further in Section 2.3.

Using this variable definition, our mathematical model is as follows. Note that in this formulation, we aggregate the facility location constraints and do not require empty balance at any of the nodes.

2.2 Model Formulation

2.2.1 Notation

Sets

R = set of composites r

T = set of truckloads t

N = set of nodes k

R_t= set of composites r which contain truckload t, $R_t \subset R$

Parameters

c_r = cost of composite r, $\forall r \in R$

F_k = fixed cost of relay point k, $\forall k \in N$

$\theta_{kr} = \begin{cases} 1 \text{ if composite } r \text{ visits relay point } k, & \forall k \in N, \forall r \in R \\ 0 \text{ otherwise} \end{cases}$

Variables

$x_r = \begin{cases} 1 \text{ if composite } r \text{ is used,} & \forall r \in R \\ 0 \text{ otherwise} \end{cases}$

$y_k = \begin{cases} 1 \text{ if a relay point is opened at location } k, & \forall k \in N \\ 0 \text{ otherwise} \end{cases}$

2.2.2 Optimization Model

$$\min \quad \sum_{r \in R} c_r x_r + \sum_{k \in N} F_k y_k$$

$$subject\ to \quad \sum_{r \in R_t} x_r = 1 \quad \forall t \in T \qquad (1)$$

$$\sum_{r \in R_k} \theta_{kr} x_r \leq |T| y_k \quad \forall k \in N \qquad (2)$$

$$x_r \in \{0,1\} \quad \forall r \in R \qquad (3)$$

$$y_k \in \{1,0\} \quad \forall k \in N \qquad (4)$$

The objective function minimizes the sum of the cost for all composites being selected (i.e., routing cost) and the fixed cost of locating RPs. Constraint (1) requires that a routing for each truckload is selected in order to satisfy the demand. Constraint (2) allows a composite to be selected only if the relay points visited by that composite are open. Constraints (3) and (4) enforce integrality for all decision variables.

2.3. Generation of Composite Variables using Templates

A challenge with this formulation is in how to generate the composite variables that are used by the model. A main contribution of this honors thesis was in developing and testing an enumeration-based approach to generate these composites. Recall that the composites represent feasible routes for truckloads. As a result, it is critical that the composites generated satisfy restrictions on operational constraints. In particular, we require each composite generated to adhere to the following constraints.

- A composite may visit no fewer than one RP and no more than three RPs
- Local distances cannot exceed a pre-specified distance limit
- Lane distances cannot exceed a pre-specified distance limit
- The total distance traveled by a load cannot exceed the given circuity allowance

To enumerate the composites for the model, we use *templates* – predefined routing patterns into which the loads fit. Each template is a different combination of nodes, relay points, local and lane distances. An example is shown in Figure 3, where a load *ij* stops at two intermediate relay points, K and L (circles represent either origin or destination nodes, and squares represent relay point). As previously discussed, we enforce all feasibility requirements explicitly when enumerating these variables.

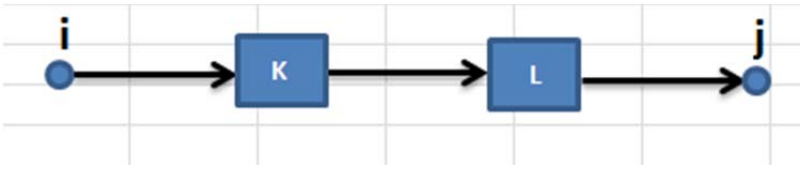


Figure 3. Template example

Using the limitations on the number of relay points visited, it is possible to create eleven different templates shown in Figure 4. Pseudo code for each of the eleven templates was created before developing the code that checks loads against potential sequences of nodes to ensure that the constraints described above are satisfied. This was then implemented in Python and used to generate all composites in the RNDP model.

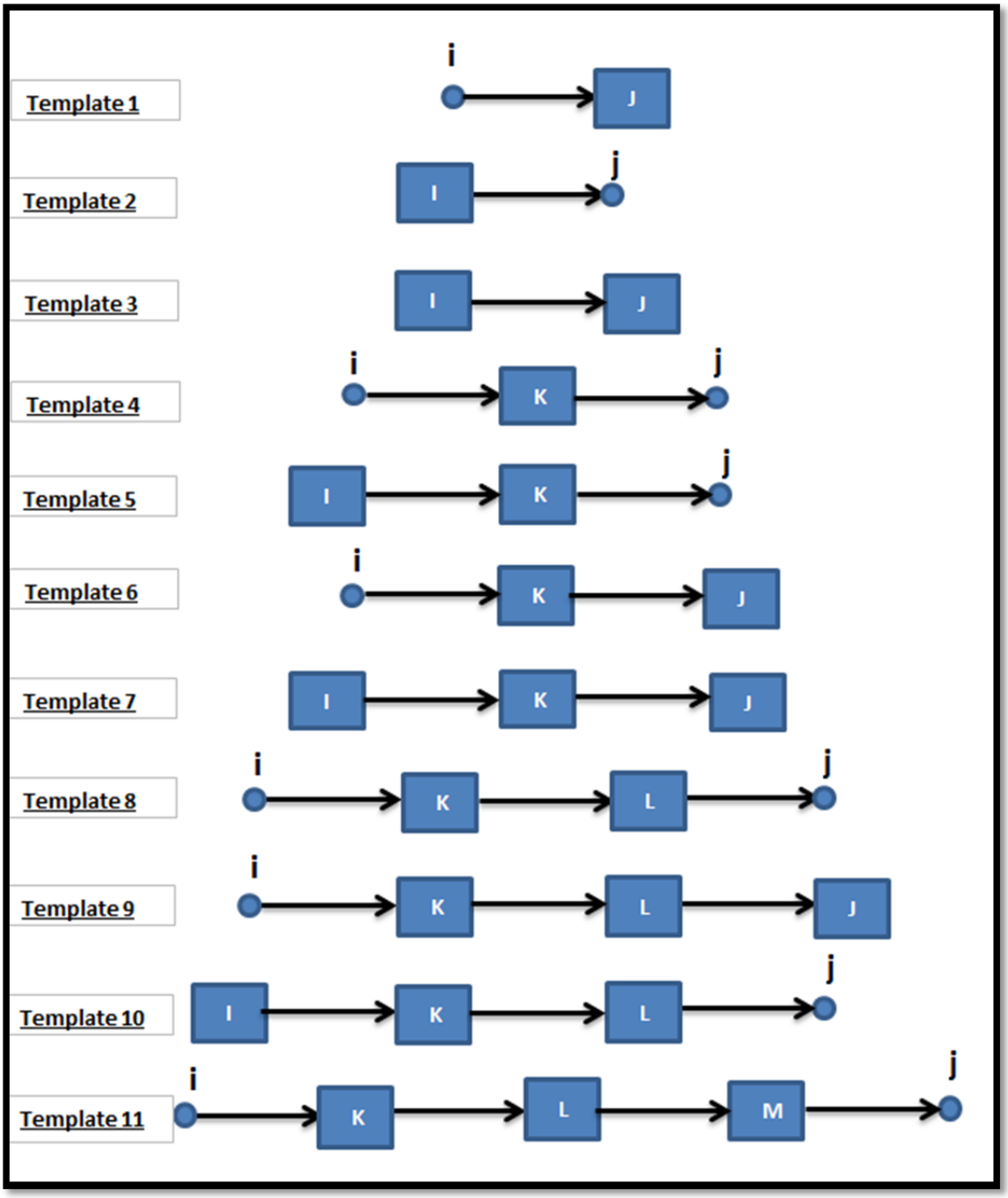


Figure 4. All template options

When the code runs, it attempts to generate each of these template types for each load for all possible combinations of intermediate nodes. However, only those which are feasible are actually enumerated and included in the model. For those templates which are feasible, the cost of these templates are calculated for inclusion in the objective function. The corresponding elements of the A matrix for each of the feasible composites is also stored so that it can be used

by CPLEX to solve the RNDP model. Note that the number of variables created for each load depends upon the characteristics of the network such as the distance of the load's origin from nearby nodes.

3. Computational Experiments

When we first tested our enumeration-based procedure to generate composites for the RNDP, it became obvious that the number of composites was the driving factor behind the time required to solve RNDP and whether or not an instance was tractable. In particular we started to realize how as the number of composites increased, the time it took to generate the composites and solve the problem increased as well. When the number of composites was extremely large (i.e., 1,000,000+), the problem would not solve as either Python or CPLEX ran out of memory.

Note that the maximum number of composites that will assume a non-zero value is the sum of number of truckloads to be shipped and the number of potential RPs. This is typically several orders of magnitude less than the number of composites generated. Using these insights, we decided to shift our focus of the research. The new research objective was to complete computational experiments that would help us identify ways that would help us reduce the number of composites without sacrificing solution quality (i.e., excluding composites that would have been used in the optimal solution). The remainder of this paper focuses on this analysis.

3.1. Dominated Templates

As we were running the code and analyzing the templates generated, it became apparent that many of the composites generated would never be used in an optimal solution. This is because they are dominated by another template type that visits the same nodes at a lower cost.

Consider the example shown in Figure 5. In this example, templates 4, 5, 6, and 7 all visit the same nodes. Assuming that the distances from i to K and K to j are less than the local distance, template 4 is dominant and composites 5, 6, and 7 should not be generated for this load, since they require more relay points to be open. However, if the distances prevent template 4 for being generated for load *ij* (e.g., the distance from i to K is greater than the local distance but less than the lane distance), then the remaining templates are evaluated to see whether they should be enumerated. Note that we checked for and eliminated all dominated template types.

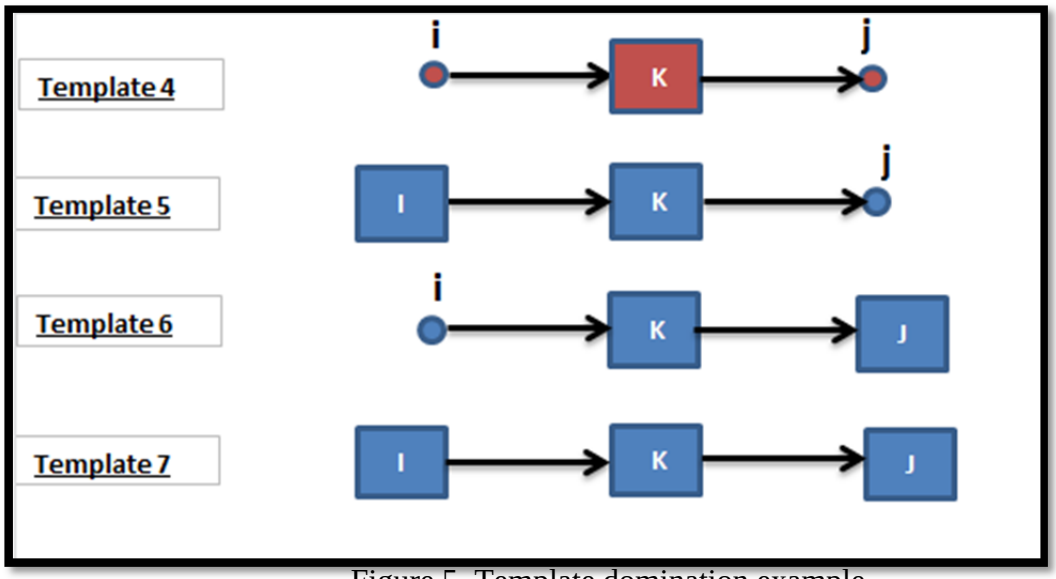


Figure 5. Template domination example.

Checking for and eliminating dominated composites allowed us to significantly reduce the number of composites generated as well as the time required to obtain the optimal solutions. For example, Table 1 shows the solution obtained for running five different replications of instances with 50 nodes and 100 loads before and after removing dominated templates.

	Number of Composites	Solution Time (sec)
No dominance reduction	52,990	391
Dominance reduction	37,765	174

Table 1. The effect of eliminating dominated templates.

Implementing dominance reduction resulted in a significant reduction in both the number of composites generated and solution time. We observed similar reductions for other instance types and problem sizes as well.

3.2. Empirical Testing

In order to determine which factors were most important determinants of run time and solution quality, we conducted an empirical analysis to of our proposed formulation. Our empirical analysis required us to generate a large number of test instances. To generate them, both the number and location of the nodes were varied. We assumed all nodes were connected in a complete network. The following table contains the different combinations of nodes and loads we tested. For each of these combinations, we generated five instances where the locations of the nodes – and hence, the underlying network – and the loads' O-D pairs varied as shown in Table 2.

Nodes	**Loads**
5	1, 5, 10
12	1, 10, 25, 50
50	1, 10, 50, 100
100	1, 10, 50

Table 2. Characteristics of test instances generated.

The x- and y-coordinates were determined by selecting random variables distributed U[0,1] and then scaling them to match the size of the area we considered (we assumed 600x600). To generate loads, we randomly selected origin-destination pairs.

4. Independent and Dependent Variables

Given problem instances to test, we wanted conduct an empirical analysis to study the impact of varying different parameters on metrics of interest. In order to do this analysis first we identified both the independent and dependent variables.

Independent Variables	**Dependent Variables**
Circuity Cost limit percentage Fixed cost Lane distance Local distance Local cost Lane cost Network size Number of nodes Number of loads	Solution time Solution cost Template IDs used Number of RPs used

Table 3. Independent and Dependent Variables Considered in our Experimental Design.

After we identified the independent variables that we could vary, we decided that we should not consider varying all these variables due both to time constraints as well as system constraints that would not allow these parameters to be varied in real life. For example, the cost of local or lane per distance unit is a variable that we can vary and analyze the values obtained; however, in real life it would not be possible to vary this value. Consequently, we decided to focus our analysis on only a subset of the independent variables identified in Table 3.

The independent variables we consider for our empirical testing were further divided into two new categories: system considerations over which we had little no control, and modeling decisions over which we had easy control. For example, although the cost of a relay point will have a huge impact on the final solution, we have no control over this. Similarly, we have no control over the number of nodes in the network, or the number of loads running through the network. Nonetheless, we wanted to see the impact of these parameters on how easily we could solve our problem, and how these parameters affected both the solution time and the cost of the

solutions we obtained for our problems.

The remaining independent variables we considered represent modeling choices or carrier preferences, and can be easily modified easily. Varying the maximum permissible circuity is an example of this type of parameter. Another example is what we refer to as the cost limit percentage above the minimum cost composite for each load which is going to be explained in Section 5.2. Once again, we ran the code with different levels for each of these independent variables to determine how they impacted both the quality of the solution and time required to obtain solutions.

The majority of the experiments performed for this thesis were run on Dell Latitude 6400 laptop with a 2.4 GHz processor and 4 GB RAM. We implemented the composite generation and the mathematical model in Python 2.6 and used CPLEX 12.2 to solve our instances. Followup experiments were run on a desktop with a 3.2GHz processor and 6 GB memory.

5.1. Impact of Network Size

As previously mentioned, the size of the network has a significant impact on the number of composites generated and consequently on the time it takes to obtain the optimal solution. By running different combinations of nodes and loads quantities, it was possible to observe an exponential growth on the composites generated. Table 4 shows how the number of composites generated and the solution time varies as the network size and number of loads varies. The numbers reported represent the averages across all five instances generated for each combination of factors.

Nodes	Loads	Number of Composites	Solution Time (sec)
5	1	84	0.620
	5	2,450	0.772
	10	9,800	0.831
12	1	714	0.407
	10	64,610	0.886
	25	364,000	0.794
	50	1,359,750	0.759
50	1	13,377	0.568
	10	1,448,510	7.444
	50	41,932,450	83.729
	100	157,922,100	307.526
100	1	131,894	1.335
	10	9,640,260	323.537
	50	273,238,350	20,563.239
	100	959,473,200	76,134.887

Table 4. Impact of instance size on number of composites and solution time.

5.2. Cost Limit Percentage

One of the independent variables on which we focused our analysis was the cost limit percentage. This is defined as a limitation above the minimum cost composite for which composites are generated for each load. For example, suppose that the minimum cost required to move a load was $100. Using a cost limit percent of .5 (i.e., 50%), composites would be generated for that load only if they were $150 or less.

To test the effect of the cost limit percentage on the solutions we obtained, we used cost limit percentage levels of 0.2, 0.3, 0.4, 0.5, 0.6, 0.7 and 1. We observed that when a cost limit percent of 1 was used, all composites were generated for all instances we considered. As a result, we did not consider higher cost savings percentages.

Since lower values of the cost limit percentage reduce the total number of composites generated, we expected to see a reduction in the solution time for lower cost limit percentages. We further expected that the higher cost composites would not be used in the optimal solution and could therefore be omitted without sacrificing solution quality. Our objective was to determine a point where the solution time is reduced, but the optimal objective function value was not affected by the cost limit percentage.

In Table 5, we show results across all instances with 50 nodes and 100 loads to illustrate the effect of the cost savings percentage on the solutions obtained. In this case, the optimal solution to the problem is obtained when a cost limit percent of .3 is use. Using this cost savings percentage, the problem solves much more quickly, requiring only 5% of that required when a cost limit percentage of 1 is used. This shows that, as we expect, the cost limit percent can significantly reduce the time required to obtain high quality solutions.

Cost Limit %	Total Number of Composites	Reduced Number of Composites	Cost	Solution time (sec)	Number RP
0.2	210,007	54,901	$ 127,146	60.10	11
0.3	210,007	97,912	$ 107,638	134.69	9
0.4	210,007	148,302	$ 107,633	652.79	9
0.5	210,007	198,722	$ 107,633	2,084.03	9
0.6	210,007	208,971	$ 107,633	2,357.56	9
0.7	210,007	209,673	$ 107,633	4,606.67	9
1	210,007	210,007	$ 107,633	3,672.14	9

Table 5. Effect of the cost limit percentage on the solutions obtained.

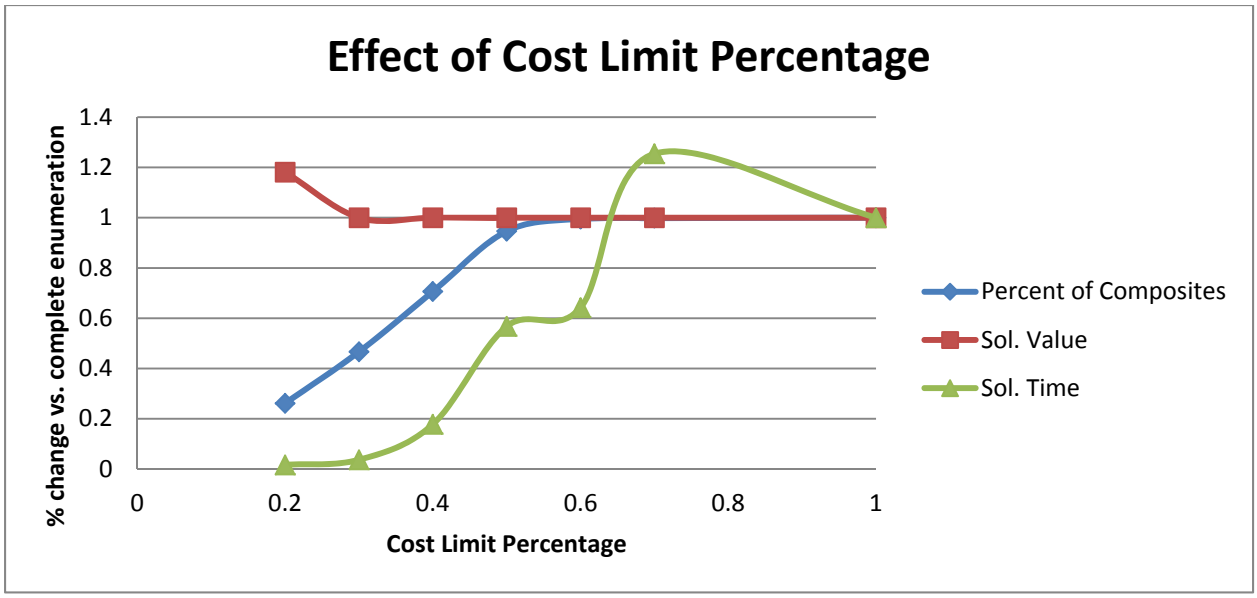


Figure 6. Effect of cost limit percentage.

However, after running significant number of instances for varying problem sizes and characteristics we observed significant variation in time required to solve the problem from one instance to another; Figure 6 shows an example where the time required to obtain the solution when the cost limit percentage is 0.7 exceeded the time required for a cost limit percentage of 1.0. Note that although they are not shown here, the shape of the curves that represent solution times (i.e., the green line) vary from one replication to another. Furthermore, some instances required very high cost limit percentages to obtain the optimal solutions. As a result, we could not draw general conclusions as to which cost limit percentage should used universally to obtain high-quality solutions more quickly; this number varies from one instance to another and therefore cannot be determined a priori.

5.3. Fixed Cost

Another independent variable we wanted to analyze was the impact of the fixed cost of opening a relay point in the solution obtained. We define the baseline fixed cost of $10,000. Given this cost, we observed across instances that approximately 50% of the cost was routing cost, and the remaining 50% was the fixed cost of installation. We wanted to understand the sensitivity of the solutions obtained to the value of the fixed cost. To do this, we added a cost multiplier to the code. This multiplier modified our baseline cost. We increased the fixed cost by factors of 2, 5, 10, and 20 to see how the solutions changed.

Multiplication Factor	Cost	Num. RP
2	$ 313,521.14	14
5	$ 733,521.14	14
10	$ 1,433,533.76	14
20	$ 2,833,540.54	14

Table 6. Effect of fixed cost on total cost and the number of RPs opened.

Surprisingly, as shown in Table 6, the model was insensitive to the cost of the relay points; the same relay points were opened regardless of their cost. This implies that the relay points are opened for feasibility reasons (i.e., to satisfy local and lane distance requirements as well as circuity constraints) as opposed to cost reasons.

5.4 Circuity

Next we analyzed the effect of varying the circuity allowance given for the maximum distance traveled. For example, suppose that the shortest path for a load *ij* was denoted by SP_{ij}. If a circuity level of 0.5 was permitted, no paths longer than 1.5 SP_{ij} could be generated.

We tested circuity limits of 0.1, 0.2, 0.3, 0.4, and 0.5. As expected, lower levels of circuity would generate fewer composites and consequently solve more quickly and vice versa. Counterintuitively as the circuity increases, the total solution cost goes down. This is because the increased routing cost is far offset by the savings that result from opening fewer relay points.

After analyzing our results, we found a few instances where the time to solve the problem decreased as the circuity increased. Consequently, we decided that it was going to be necessary to reproduce a second set of experiments generating a significant number of replications to determine how frequently this kind of exception occurs and if it was going to affect our subsequent analysis. These results are discussed in Section 5.4.2

The second set of experiments was reproduced on the faster computer described previously. We generated 30 instances each for networks with 50 nodes, 80 nodes and 100 and 500 loads. By running these experiments we wanted to analyze and explore relative impact of circuity on solution time.

As a result of running all these new instances, our first observation was that increasing the circuity value led to a decrease in the number of relay points opened as shown in Table 7. Consequently, the total fixed cost decreased.

		Circuity				
Nodes	**Loads**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**
50	100	23	18	16	14	14
	500	35	29	25	23	22
80	100	23	18	16	15	15
	500	40	31	28	26	25

Table 7. Average number of relay points opened as a function of circuity.

This led, however, to loads taking routes with additional circuity and, consequently, a higher total routing cost. In Table 8 it is possible to observe the increase on the average routing cost.

		Circuity				
Nodes	**Loads**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**
50	100	$ 29,008.09	$ 29,383.26	$ 29,821.79	$ 30,274.67	$ 30,440.09
	500	$ 139,974.26	$ 141,179.05	$ 142,574.30	$ 143,675.34	$ 144,570.55
80	100	$ 27,241.46	$ 27,756.05	$ 28,174.91	$ 28,510.63	$ 28,668.08
	500	$ 134,008.55	$ 135,772.60	$ 136,991.92	$ 138,051.00	$ 138,642.10

Table 8. Average routing cost as a function of circuity.

The increase in the permissible circuity allowed the program to generate more composites (see Table 9) and to obtain a better overall solution. However, this resulted in a higher time to solve the problem (see Table 10).

		Circuity				
Nodes	Loads	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**
50	100	4,297	10,953	20,031	31,161	43,763
	500	20,943	52,859	96,484	150,176	211,025
80	100	12,213	33,568	63,690	101,528	145,558
	500	61,826	169,633	321,683	513,226	732,298

Table 9.Average number of composites as a function of circuity

		Circuity				
Nodes	Loads	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**
50	100	0.1821	0.2475	0.4374	0.6786	1.1120
	500	0.3397	1.1298	2.8188	5.1028	8.7351
80	100	0.5276	1.2367	3.6633	13.1562	59.0214
	500	2.6337	13.8220	32.3391	90.2927	439.3216

Table 10. Average solution time (sec) as a function of circuity

After analyzing our results, we created graphs which illustrate the tradeoff between solution time and cost. These are shown in Figure 7 for our instances of 100 and 500 loads on networks of 50 and 80 nodes.

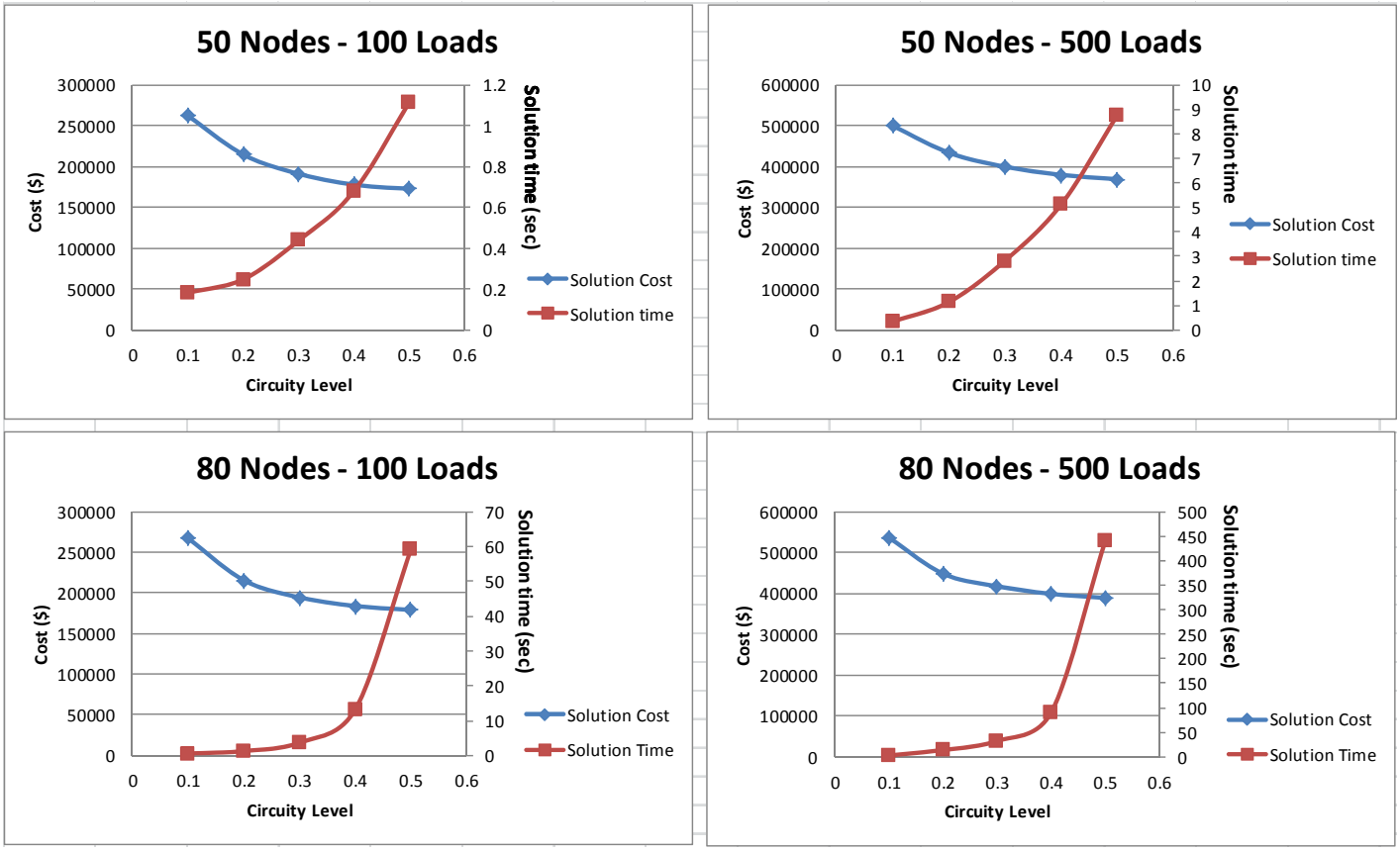


Figure 7. Effect of circuity level on cost and solution time.

These graphs show the dramatic effect that the circuity has on the time required to obtain the optimal solutions. Additionally, they show that beyond a certain level of circuity, the solution quality is affected only slightly whereas the effect on the solution time was significant. However, in comparing instances it was not possible to obtain a consistent cutoff for the level of circuity, although a value between 0.3 and 0.5 appears to be appropriate.

6. Templates analysis

Another strategy we considered to reduce the number of composites generated by the model was to analyze which templates were actually used in the optimal solution. We observed that while some of the templates were used a significant number of times, others were barely used or even not used at all. Table 11 shows the number of composites generated for each template type when running instances for 50 nodes and 100 and 500 loads across the 30 different replications.

		Template										
	Circuity	1	2	3	4	5	6	7	8	9	10	11
100 Loads	0.1	193	430	636	1,532	820	2,304	-	11,584	9	-	-
	0.2	201	364	399	1,760	815	2,040	14	12,312	-	-	-
	0.3	196	334	354	1,824	885	1,998	7	12,416	-	-	-
	0.4	169	332	333	1,924	860	1,938	-	12,624	-	-	-
	0.5	164	322	339	1,896	810	1,938	-	12,816	9	-	-
500 Loads	0.1	1,206	1,970	2,775	8,524	2,990	7,884	21	62,704	-	-	-
	0.2	967	2,150	1,893	9,580	2,730	7,080	14	65,608	27	-	-
	0.3	891	2,014	1,785	10,128	3,080	7,464	14	64,888	18	-	-
	0.4	824	1,892	1,587	10,340	3,435	8,004	35	64,688	36	-	-
	0.5	749	1,832	1,638	10,404	3,725	8,526	28	64,080	72	-	-

Table 11. Average number of templates generated for different circuity levels.

Table 11 shows that even if we increase the number of loads, there are still templates not being utilized at all. For example for some circuity levels templates 7 and 9 are used only when our instances have 500 loads and, even then, very little. Table 12 show the percentage of use for each template for the same instances with 100 and 500 loads.

		Template										
	Circuity	1	2	3	4	5	6	7	8	9	10	11
100 Loads	0.1	1.10%	2.46%	3.63%	8.75%	4.68%	13.16%	0.00%	66.16%	0.05%	0.00%	0.00%
	0.2	1.12%	2.03%	2.23%	9.83%	4.55%	11.39%	0.08%	68.70%	0.00%	0.00%	0.00%
	0.3	1.09%	1.85%	1.97%	10.13%	4.91%	11.09%	0.04%	68.92%	0.00%	0.00%	0.00%
	0.4	0.93%	1.83%	1.83%	10.58%	4.73%	10.66%	0.00%	69.44%	0.00%	0.00%	0.00%
	0.5	0.90%	1.76%	1.85%	10.36%	4.43%	10.59%	0.00%	70.06%	0.05%	0.00%	0.00%
500 Loads	0.1	1.37%	2.24%	3.15%	9.68%	3.39%	8.95%	0.02%	71.19%	0.00%	0.00%	0.00%
	0.2	1.07%	2.39%	2.10%	10.64%	3.03%	7.86%	0.02%	72.86%	0.03%	0.00%	0.00%
	0.3	0.99%	2.23%	1.98%	11.22%	3.41%	8.27%	0.02%	71.87%	0.02%	0.00%	0.00%
	0.4	0.91%	2.08%	1.75%	11.38%	3.78%	8.81%	0.04%	71.21%	0.04%	0.00%	0.00%
	0.5	0.82%	2.01%	1.80%	11.43%	4.09%	9.36%	0.03%	70.38%	0.08%	0.00%	0.00%

Table 12. Templates average percentage utilization for different circuity levels

By analyzing the different percentages of utilization it is possible to notice that regarding the number of loads used, there are templates with an insignificant usage or no usage at all. On the other hand it is possible to see that template 8 has an average of utilization of about 70% for every circuity level. Consequently, it was possible to define that omitting the use of some templates when running the code would help us to reduce the total number of composites generated without significantly impacting the solution quality.

7. Conclusions

The objective of this honors thesis was to analyze ways to reduce the number of composite variables generated to solve the RNDP, and to study the effect of these reductions on the quality of solutions obtained. To do this, we conducted an empirical analysis to determine how different parameter values impacted various performance measures such as the cost of the solutions obtained and the time required to obtain the solutions. This analysis uncovered several interesting insights, and promising strategies that could be used to reduce the number of

composites generated and, consequently the time required to solve the RNDP, without sacrificing solution quality.

Our analysis reveals that network size (i.e., more nodes and more loads) has a dramatic impact on the time required to solve the RNDP. As the number of loads and/or nodes increases, the time it takes to solve the problem increases exponentially. Our formulation is intractable when instances are too large due to a memory shortage. Note that this occurs during the build stage. This memory shortage is due to the large number of composites generated, not due to the size of the branch and bound tree used to solve the problems. Another insight that results from our analysis is that varying the fixed cost has no effect on the number of relay points opened or the location of those relay points.

Overall we were able to determine that there some ways to limit the number of composites generated are better than others. For example, as a result of this research we observe that varying the circuity level is a more effective strategy to limiting the number of composites than varying the cost limit percentage.

8. Future work

There are a number of promising research directions that should be pursued to further reduce the number of composite variables which are generated. In particular, an exact approach such as column generation, where we avoid generating unused variables, could be used to determine whether or not composites should be generated. A further way to explore the reduction of composites would be the implementation of a process where composites are generated based on specific templates determined by the distance between origin and destination for the loads (i.e. if the distance doesn't exceed a given threshold, only use templates with up to two relay

points). Another interesting area for future research lies in understanding the variability from one instance to another. Our results showed significant variation from one instance to another when analyzing the effect of the cost limit percent on total solution time. It would be interesting to understand why this variance occurs.

References

American Trucking Associations (2008). *Trucking Activity Report.* Retrieved from: http://www.truckline.com/.

Campbell, J.F. (2005). Strategic Network Design for Motor Carriers. In A. Lagevin and D. Riopel (Eds.), Logistics Systems-Design and Optimization (245-278). New York, NY: Springer-Verlag.

Morris, Frank. (2011). *Thousands Of Trucking Jobs, But Few Take The Wheel*. Retrieved from: http://www.npr.org/2011/10/31/141791952/thousands-of-trucking-jobs-but-few-take-the-wheel

Taylor, G.D., Whicker, G.L. and Usher, J.S. (2001). Multi-Zone Dispatching in Truckload Trucking. *Transportation Research: Part E,* 37, 375-390.

U.S. Bureau of Transportation Statistics (2004). *United States: 2002 Commodity Flow Survey.* Retrieved from: http://www.brs.gov/publications/commodity_flow_survey/.

U.S. Bureau of Transportation Statistics (2009). 2007 *CFS: Final National Tables-December 2009.* Retrieved from: http://www.bts.gov/publications/commodity_flow_survey/.

Üster, H. and Maheshwari, N. (2007). Strategic Network Design for Multi-Zone Truckload Shipments. *IIE Transactions,*39, 177-189.

Vergara Hector. (2010). *Optimization Models and Algorithms for Truckload Relay Network Design*