Reduction of Variables for Solving Logistic Flow Problems

K. Kalicharan[1, 2], F. Phillipson[2], A. Sangers[2], M. De Juncker[2,3]

1. Delft University of Technology, Delft, The Netherlands
2. TNO, The Hague, The Netherlands
3. Eindhoven University of Technology, Eindhoven, The Netherlands

Corresponding author: frank.phillipson@tno.nl

Abstract: *In logistic problems, an Integral Multi-Commodity Network Design Problem on a time-space network is often used to model the problem of routing transportation means and assigning freight units to those means. In Physical Internet and Synchromodal networks an interactive planning approach is preferable, meaning that calculation times of a single planning step should be short. In this paper we provide finding ways to reduce the number of variables in the problem formulation, that are effective at reducing the computation time for ILP-based solution methods.*

Keywords: *Logistic Space Time Network, Logistic Flow Routing, Synchromodality, Variables Reduction Techniques*

1 Introduction

In this research paper we look at a transportation system where logistic units (e.g., containers) travel freely through this network. Decision makers here can be logistic service providers or clients controlling the stream of their containers, or intelligent containers or other smart logistic units themselves. This occurs in Physical Internet and in Synchromodal or Intermodal networks.

These types of transport networks allow for many different options of transportation. A unit can be trucked from its origin to its destination, but (part of) the route can also be done by barge, plane or train. In synchro- and intermodal networks, a logistics service provider (LSP) is responsible for managing the flow of containers from their origins to their destinations. Especially in large logistics networks, LSPs may require assistance from algorithms to make good routing choices. These algorithms can be used to reach economic and emission-reduction targets by offering decision support for optimizing the (intermodal) transport chain (Janic 2007). In Physical Internet and Synchromodal networks, a more interactive planning scheme is preferable, following the dynamic and uncertain nature of the underlying networks. Here routing decisions can (or have to) be made, each time logistic units arrive at an intermediate node, incorporating the decisions of other agents and, possibly, the uncertainty of decisions or events in the future.

In logistic (service) network planning problems Space-Time Networks (STN) are often used for the representation (De Juncker et al., 2017). On this STN a non-negative integral minimum cost multi-commodity flow problem (MCMCF) is solved to get the overall optimal solution (Crainic, 2000). Here, all terminals are modelled as nodes. Services go from these terminals to other terminals in a certain amount of time. These services are modelled by arcs. To solve these problems, constraints can

be relaxed to obtain a simpler problem. This yields lower bounds that together with heuristically found upper bounds can be combined in for instance a branch-and-bound algorithm, see, among others, the paper by Crainic et al. (2001) and Holmberg and Yuan (2000). Holmberg and Hellstrand (1998) propose an exact solution method for the uncapacitated problem based on a Lagrangian heuristic. A dual ascent procedure is treated by Balakrishnan et al. (1989), which finds lower bounds within 1 − 4% of optimality. Heuristics and meta-heuristics (such as Tabu Search, Simulated Annealing and Genetic Algorithms) are also widely used. See for instance the paper by Crainic et al. (2000), who look at a path-based formulation of the same problem and solve it with tabu search. Other papers looking into these meta-heuristics are among others: Bai et al. (2012), Chouman and Crainic (2012) and Pedersen et al. (2009).

To allow for a more interactive use of this solution direction, short calculation times are crucial. In this paper we look at the problem of scheduling services and assigning transportation units to it. We start with modelling this problem as an Integral Multi-Commodity Network Design (MCND) problem. We will present novel reduction approaches to reduce the computation time when solving the problem. The MCND problem will be introduced in Section 2. In Section 3 the proposed reduction approaches are introduced, using the specific structure of the problem. Next, in Section 4 computational results will be presented and we will end with conclusions in Section 5.

2 Multi-Commodity Network Design Problem

The main goal of this paper is about efficiently and simultaneously routing transport means and scheduling transportation units. Without losing any generality, we will say vehicle if we mean a transportation mean and container if we mean the transportation unit. One of the models that could be used here is the Capacitated Fixed Charge Network Flow Problem from Ghamlouchee et al. (2004), Hewitt et al. (2010), Magnanti and Wong (1984) and Rodríguez-Martín and Salazar-González (2010). We have a directed graph $G = (V, E)$ (or multigraph) that contains all the nodes and arcs in the network. We have a set of commodities K that represent the bookings/orders. Every commodity $k \in K$ has, without loss of generality, one source node s_k and one sink node t_k. The parameters c_e, for $e \in E$, are the capacities of the arcs. The parameters $f_{e,k}$, for $e \in E$, $k \in K$, determine the per unit cost of commodity k on arc e. The parameters $d_{v,k} = d_k$ if $v = s_k$, $d_{v,k} = -d_k$ if $v = t_k$ and $d_{v,k} = 0$ otherwise, where d_k is the demand/size of commodity k. The variables $x_{e,k}$ depict the magnitude of the flow of commodity k on arc e. Finally, we define the design variables y_e, $\forall e \in E$, that are one if the service at link e is active and zero otherwise. In intermodal transport these design variables are normally one if and only if the corresponding vehicle travels the corresponding link. Many possible arcs to travel are added for a vehicle and after the optimization process, it is decided which design variables are one; ergo, which routes the vehicles should travel. The graphs for these models are often time-space networks, but other graphs can also be used (Sharypova, 2014). The optimization problem now looks like:

$$\min \sum_{k\in K} \sum_{e\in E} f_{e,k} x_{e,k} + \sum_{e\in E} g_e y_e \quad (1)$$

$$s.t. \sum_{e\in\delta^+(v)} x_{e,k} - \sum_{e\in\delta^-(v)} x_{e,k} = d_{v,k} \qquad \forall\, v \in V, \forall\, k \in K \quad (2)$$

$$\sum_{k\in K} x_{e,k} \le c_e y_e \qquad \forall\, e \in E \quad (3)$$

$$x_{e,k} \ge 0, y_e \in \{0,1\} \qquad \forall\, e \in E, k \in K \quad (4)$$

The first part of the objective function minimizes the cost using the $f_{e,k}$. The second part is a link cost, if a vehicle travels a certain link $e \in E$, then a certain fixed cost g_e is added. The flow conservation constraints (2) make sure that the total amount of a commodity that enters the node also leaves the node, except for the sources and sinks. The capacity constraints (3) say that if an arc in the network is not travelled by a vehicle, then no commodity flow may be on that arc. If a vehicle does travel an arc, then the container flow on that arc can be at most the capacity of the edge.

This model is in some papers, see Chouman and Crainic (2012), Pedersen et al. (2009), Vu et al. (2012) and Vu et al. (2014), extended to include what are called design-balanced constraints:

$$\sum_{e\in\delta^+(v)} y_e - \sum_{e\in\delta^-(v)} y_e = 0 \;\forall\, v \in V \qquad (5)$$

These constraints make sure that everywhere a vehicle arrives, it also leaves. This means that the routes for the vehicles are directed cycles. So the network we use, should contain directed cycles. When working over a time-space network, an additional arc should be added from the sink of vehicle type w of set W to the source of vehicle type w to make sure it is possible to have a directed cycle.

In Sharypova (2014) a similar continuous time ILP (Integer Linear Programming problem) is proposed. This model also has time variables and some more types of constraints. An extension of the design-balanced service network design problem is given in Li et al. (2017). The model takes into account the usage of vehicles and the opening of corridors. In Andersen et al. (2009) an other extension is derived and in Joborn et al. (2004) a model that shares some resemblances is applied to freight car distribution in scheduled railways. A completely different ILP that can handle the same sort of problem is given in Huizing (2017).

The problem we will propose is similar, though contains a few key differences. First, We will consider two vehicle types: not flexible, high capacity, low cost vehicles that need to be scheduled in advance (barges, train etc.) and flexible, low capacity, high cost vehicles (trucks, transporters, etc.). From here we will refer to the first type as barge and to the second type as truck. Second, only the first part of the objective function of the service network design problem is used. Third, the flow conservation constraints for the vehicles are slightly different than the design-balanced constraints: in our model the number of non-truck vehicles is pre-specified and only for the non-truck vehicles, vehicle flow conservation constraints are added. We call our problem the Integral Multi-Commodity Network Design (MCND) Problem. The model uses a time-space network, wherein the routes of the vehicles are not known in advance. The arcs in the time-space network are possible links that a vehicle can travel. For the trucks we add an arc for every time stamp from every terminal to every terminal. Naturally, the travel time is taken into account. We repeat this process for the first barge, second barge etc. Arcs corresponding to different vehicle (types) that run between the same time-space nodes are not merged. This way it is assured that all the links of all the possible routes they can take are included in the time-space network. The Integral MCND problem then is:

$$\min \sum_{k\in K} \sum_{e\in E} f_{e,k} x_{e,k} \qquad (6)$$

$$\text{s.t. } \sum_{e\in\delta^+(v)} x_{e,k} - \sum_{e\in\delta^-(v)} x_{e,k} = d_{v,k} \qquad \forall\, v \in V, \forall\, k \in K \qquad (7)$$

$$\sum_{e\in\delta^+(v)\cap E_w} y_e - \sum_{e\in\delta^-(v)\cap E_w} y_e = b_{v,w} \qquad \forall\, v \in V, \forall\, w \in W\backslash\{truck\} \qquad (8)$$

$$\sum_{k\in K} x_{e,k} \le c_e y_e \qquad \forall\, e \in E\backslash E_{truck} \qquad (9)$$

$$\sum_{k\in K} x_{e,k} \le c_e \qquad \forall\, e \in E_{truck} \qquad (10)$$

$$x_{e,k} \geq 0, y_e \in \{0,1\} \qquad \forall\, e \in E, k \in K \qquad (11)$$

For all non-truck arcs e in the network, we have created a discrete design variable determining if the arc is used (11). The y_e are binary variables. The container flows are modelled with the variables $x_{e,k}$ and still have to be integral and non-negative (11). We assume that trucks do not necessarily need to be used the whole day, whereas barges do have to be used the whole day.

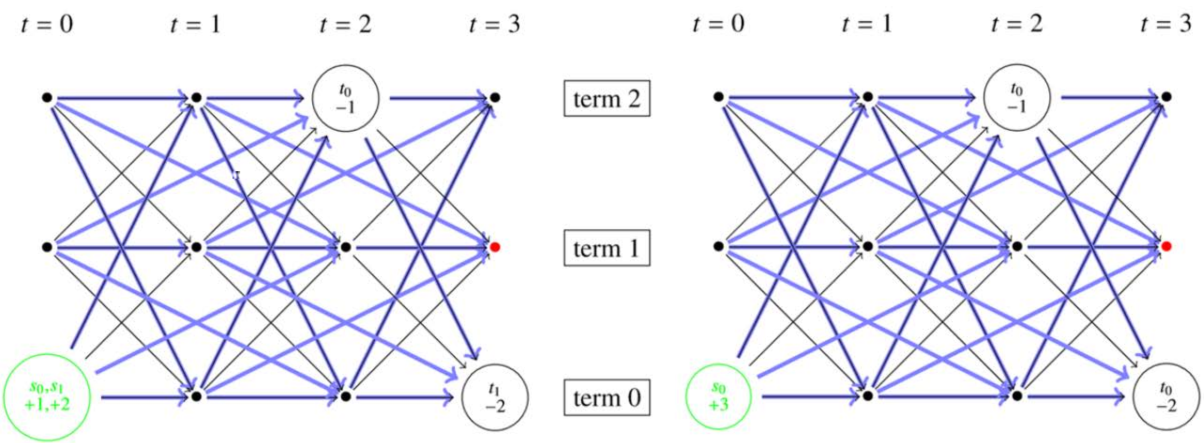

Figure 1: Combined bookings shared source.

For the paths of the barges to make sense we should add constraints that disallow the barge to teleport or travel multiple links at the same time. Flow conservation constraints for $w \in W\backslash\{\text{truck}\}$ (8) can do exactly this in the same the flow conservation constraints (7) work for the commodities. In these constraints we have $b_{v,w}$ which describe the time-space nodes that are the sink and source for a $w \in W \setminus \{\text{truck}\}$. In more detail for such w we have that $b_{v,w}$ equals b_w if v is the source node of w, equals $-b_w$ if v is the sink node of w and 0 otherwise, where b_w is the number of vehicles of type $w \in W \setminus \{\text{truck}\}$. This is normally 1 unless the vehicle reduction has been applied. The capacity of an arc is dependent on the number of vehicles that travel the arc (9). For the truck arcs we have different capacity constraints (10). Normally a vehicle has the same capacity the whole day. So we can then replace the c_e in the capacity constraints for the arcs, by capacity constants for the vehicle, c_w. For the trucks, however, c_e is the maximum number of trucks that can be deployed on link $e \in E_{\text{truck}}$. We assume a truck can carry exactly one container. Thus, the number of trucks that travel an arc $e \in E_{\text{truck}}$ is equal to the number of containers that are trucked on that arc $\sum_{k \in K} x_{e,k}$.

3 Variable Reductions

Reducing the number of variables of an ILP may reduce the computation time required to solve it. For that reason we will look at several ways to remove variables from the MCND ILP, as proposed in the previous section. A simple approach is to arbitrarily remove arc variables. However, then we might remove arcs that would be used in an optimal solution of the original problem. So in this section we try to introduce variable reductions in a smart way that do not change the optimal solution value too much. Note that the used variables are indexed over the locations, time stamps, vehicles and commodities in the MCND problem. Reducing the size of those sets will reduce the number of variables. We will present the reduction in the next subsections, ordered using the set they reduce.

3.1 Commodity Reductions

Reduction A: Same Sink/Source In the model we assume all the containers in one booking combined in one commodity. This way there are less variables than if all containers would be

a separate commodity. It is however possible to reduce the number of commodities even more if the following condition holds. If multiple bookings have the same sink, then these bookings can be combined into one commodity, as shown in Babonneau et al (2006). This can be done also if they share the same source, see Figure 1. In that figure, the values of $d_{v,k}$ are visualized for the sink and source nodes of the commodities. In the left figure, a commodity is a booking, in the right figure a commodity is two bookings.
The problem with combining them if they have different sinks and sources, is that a container of booking 0 can be transported to the sink of booking 1, if they are put in the same commodity.
If two bookings o_0 and o_1 have similar sinks for example if they have to be transported to the same destination location, then the same reduction is possible. If o_0 has to be there one time stamp earlier, then we can set the sink of o_1 to the same sink as that of o_0. Note that by doing this the optimal solution may become worse. After this we combine them in a single commodity.

Reduction B: Disjoint Time Frame Bookings In the ILP, arc variables are defined for a booking for every vehicle arc in the time-space network. Even those that start before the booking is released or end past its deadline. Suppose we have two commodities of which one has its deadline before the release time of the other one. Note that these bookings can be combined by putting them together in one commodity.
We can combine bookings in a greedy way: We start with the first one that is released and we add to the same commodity the first booking that is available after its deadline. We repeat this until it is no longer possible to add more bookings to the same commodity. After which we repeat this process for the next commodity. Clearly these bookings in the same commodity will not use the same arcs because there is no time stamp for which they are simultaneously available in the network. Every booking is available during a certain time frame.
Theorem 1. *This greedy algorithm finds an optimal way to combine the bookings, that is, minimizing the number of commodities, such that their time frames do not overlap.*

Proof. For readability we assume that none of the release times are equal, however, the proof can be extended for cases where there are bookings with equal release times. Suppose we have an allocation that minimizes the number of commodities by combining bookings in a certain way. In that allocation we start with examining the first booking b_{i_1} that is released. If the next (in time) booking in the same commodity, b_{i_2}, is not the first one released after the first booking, b_{i_1}, then we swap the first booking released after it; b_{i_3} and everything in the same commodity as b_{i_3} later in time, with b_{i_2} and everything released after b_{i_2} on the same commodity as b_{i_2}. Then we repeat this process for b_{i_3} etc. Until we are done for the commodity and then we move to the first booking released that is not on a commodity that we already handled and do the same for that commodity, but we do not move the bookings that are on a commodity that is already ’done’. The solution remains feasible and the number of commodities that are used does not increase. When we are done with all the commodities, we have found an allocation that is found by the greedy algorithm. □

3.2 Vehicle Reductions

Reduction C: Same Vehicle Type In the MCND problem we have a set of vehicles W = {truck,barge0,barge1,... }. The trucks are already combined in the model, it is also possible to combine the barges in the model assuming they all have the same travel times and capacities. So then we get W = {truck, barge}. If there are barges of types A and B we take W = {truck, typeAbarge ,typeBbarge}. In the MCND model the y_e variables are not binary variables

anymore, but more general discrete variables. They keep track of the number of barges that take arc *e*. In the capacity constraints these variables are multiplied with the capacities per barge to model the total barge capacity for a certain link.
If the barges are modelled individually, then for every barge a source and sink has to be given. With the reduction, it is possible to add multiple sources and sinks. So the barges still have the freedom to start from or end at different locations. Though, we can only specify the number of barges that has to arrive at a certain sink. It is not specify which individual barge has to arrive there, if there are multiple sinks. Furthermore, if too many sinks and sources are added, the number or possible paths for the barges might increase too much, also increasing the size of the feasible region.
An advantage of bundling different vehicles in the model together into a single vehicle type, is that this is a way to avoid problems with symmetry and reduce the number of variables in the model. If the individual vehicles are modelled separately, many different solutions that are equivalent in practice have different variable allocations in the model. For instance, if there are ten identical barges that start at location 0 and one container needs to be moved from location 0 to location 1, then barge 0 can transport the container or barge 1 can transport it, etcetera. As the barges are identical, these solutions are equivalent in practice. If these vehicles are put together in one index *w* in the model, then the container is not allocated to a specific barge in the model. That has to be done in a post-processing step.

3.3 Arc Reductions

Reduction D: Source/Sink Location In this reduction we use the property that if some part of the route the commodity needs to be trucked, then it suffices to do that as soon as it is possible to do so. We also use the fact that it is always shorter to truck directly to a location, than through another location.

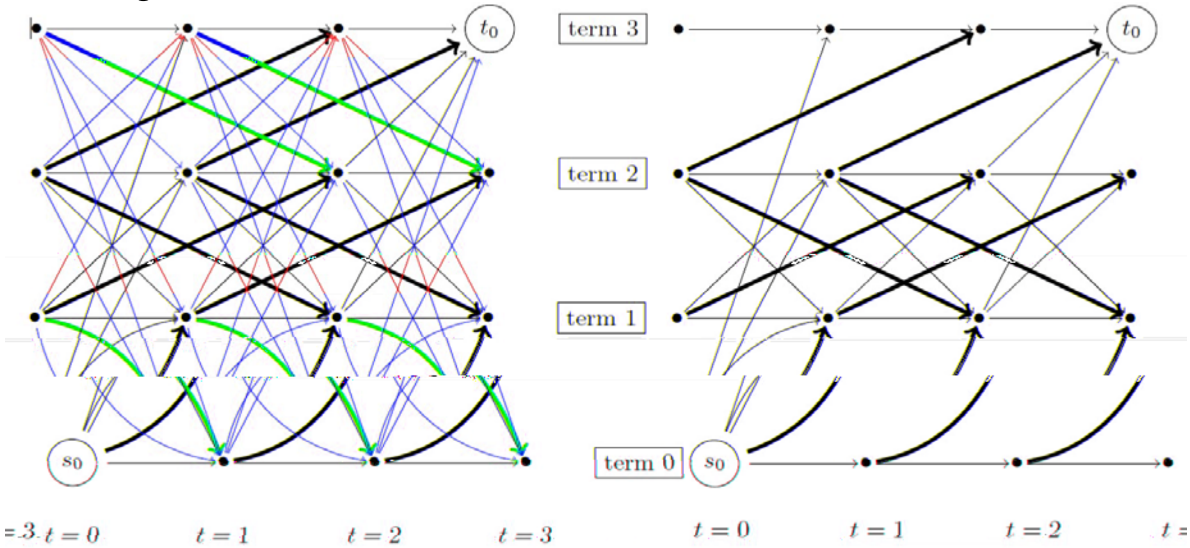

Figure 2: Reduction D for MCND.

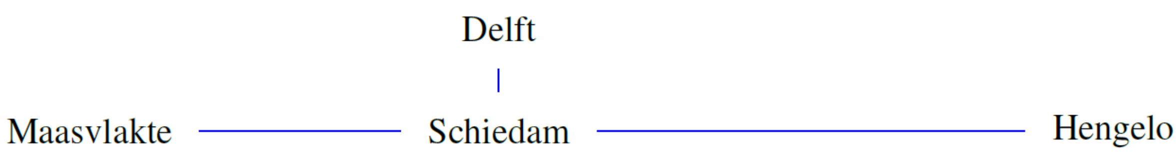

Figure 3: Waterway Connections

In the MCND problem on a time-space network some non-horizontal truck arc variables adjacent to a source location of a booking can be removed. It suffices to only add truck arc variables for a commodity from its origin location to every other location at its release time. The other non-horizontal truck arc variables adjacent to the source location can be removed. Similar things can be done for its sink location, though the arc from the source location to the sink location at the release time is never removed. Additionally, non-truck vehicle arc variables that go to the source location or leave from the sink location can be removed. In Figure 2 we apply reduction D. The truck arcs are the thin arcs, the barge arcs the thick arcs. The truck arc variables that are removed for booking 0 are in red and the barge arc variables that are removed in blue.

Reduction E: Obsolete Barge Links Arcs in the time-space network that can never be travelled by some barge, because they start before the time the barge can be at the location, are removed. These links can never be taken by any container. Similarly barge arcs can be removed that depart too late at a location.

3.4 Location Reductions

Reduction F: Minimal Paths We call a (*loc*1,*loc*2) path in a network minimal if the path has no sub path that is a (*loc*1,*loc*2) path. For every commodity k we have that every path that is not a minimal (s_k,t_k) path in the space network can be removed. In our model we can use that every location that is not on a minimal (s_k,t_k) path in the space network can be removed for commodity k. In Figure 3 we see a space network with the waterway connections of several locations. Let $k \in K$ be a commodity with source location $\chi(s_k)$ =Maasvlakte and sink location $\chi(t_k)$ =Hengelo, then we can conclude that arc variables that correspond to links that go to/from Delft do not have to be added for commodity k, if this reduction is applied.

Reduction G: Direct Connection The network of the locations, waterways and roads can have a specific, recognisable, structure. This structure can be used. If shipping from location *loc*0 to location *loc*2 means shipping through location *loc*1. Then no arcs from *loc*0 to *loc*2 have to be added. It suffices to have arcs from *loc*0 to *loc*1 and from *loc*1 to *loc*2. See Figure 4 for an example, there the red barge arcs are removed because of the structure of the waterways. We recommend taking a dense time grid with this reduction, because larger time steps may adversely affect the accuracy of the travel times between certain locations.

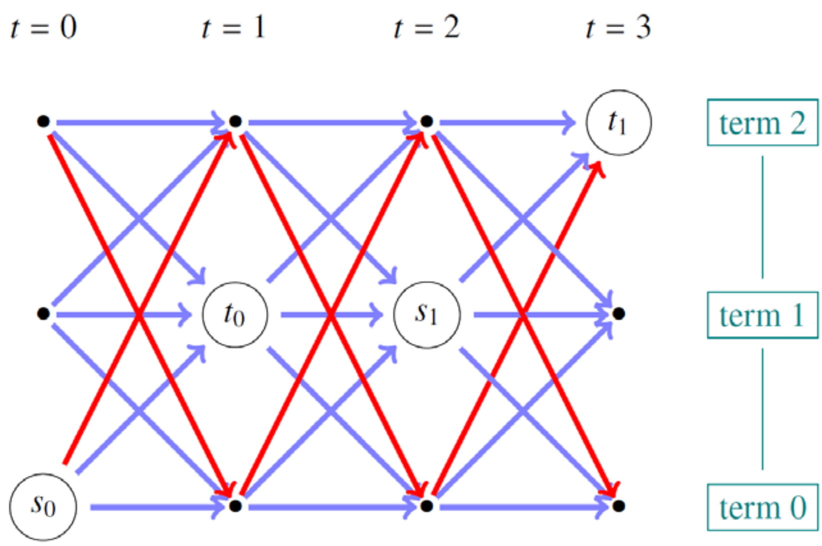

Figure 4: Direct Connection Reduction.

Reduction H: Locations In-between Every commodity k has an origin location $\chi(s_k)$ and a destination location $\chi(t_k)$. Let $d(\chi(s_k),\chi(t_k))$ be the Euclidean distance between $\chi(s_k)$ and $\chi(t_k)$, then we set all variables going to or from a location loc with $d(\chi(s_k),loc) > d(\chi(s_k),\chi(t_k)) + \delta$ and/or $d(\chi(t_k),loc) > d(\chi(s_k),\chi(t_k)) + \delta$ to zero. The δ should be chosen large enough to include locations that could be interesting for commodity k. Instead of the distance as the crow flies, it is also possible to use the trucking time for every pair of locations. This reduction can cut away optimal solutions. For example, in some problems, first trucking a container further away from your destination before shipping it to the destination, would have given the optimal solution.

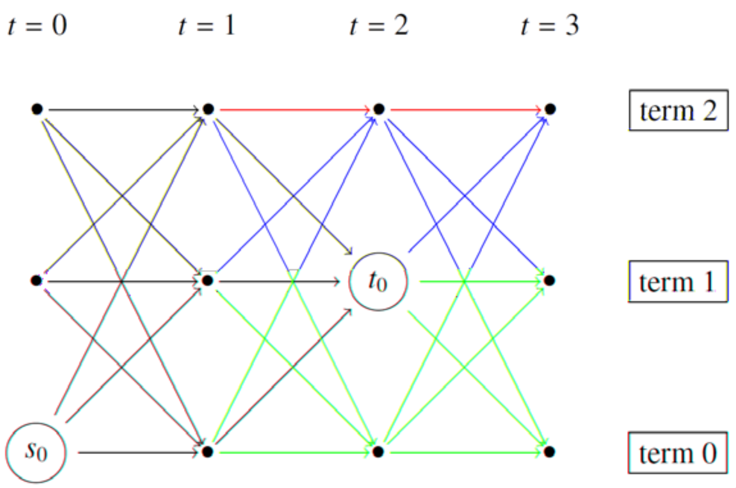

Figure 5: Obsolete Time Reduction

3.5 Time Reductions

Reduction I: Obsolete Time Reduction Let $v \in V$ be a time-space node, then we define $\tau(v)$ to be its *time* and $\chi(v)$ its *location*. Clearly a commodity can never take arcs that begin before its origin node time or end after its deadline. Instead of combining bookings in a commodity as in reduction B, one could remove those obsolete arcs entirely from the model. For every commodity $k \in K$ we remove all variables with $t + \text{a}(loc1,loc2,w)$ bigger than the time of its sink node $\tau(t_k)$, so $t + \text{a}(loc1,loc2,w) > \tau(t_k)$. We also remove all arcs with $t < \tau(s_k)$.

This reduction can even be enhanced by also removing variables with: $t + \text{a}(loc1,loc2,w) = \tau(t_k)$ and $loc2$, $\chi(t_k)$, where $\text{a}(loc1,loc2,w)$ is the travel time for vehicle (type) w from location $loc1$ to location $loc2$. Similarly for the source we obtain $t = \tau(s_k)$ and $loc2$, $\chi(s_k)$.

In Figure 5 we see an example of the truck arc variables the basic reduction removes for commodity 0 in red.

Reduction J: Time Slot Reduction It is not always possible to go to a terminal. If the available time slots are known, then that information can be used to severely reduce the number of variables in the model. If an arc goes to a time-space node for which the location is not available at that time, then we instead let the arc go to the time-space node at that location that corresponds to the soonest time at which the location is available. If no such time exists we remove the arc completely. If we get parallel arcs that correspond to the same vehicle type, then we merge them. We do a similar process for if the time-space node from which the arc leaves is not an available time-space pair.

4 Results

In this section we present results of the reduction in calculation time by using some of the proposed network reductions on a test instance. The instance has eight terminals locations and two groups of six barges. All barges that belong to the same group have the same capacities, travel times, and begin and end location. Therefore, all barges in the same group are modelled with one variable $w \in W$ in the model (Reduction C) unless stated otherwise. We assume an infinite number of trucks at every terminal, but restrictions may be added if that is desired. The travel times of the vehicles are based on data from practice and the truck travel times on data from Google Maps (Page and Pichai, 2018). Every truck can carry exactly one (40 feet) container. We choose to look at a time span of 36 hours with time steps of one hour, where 50 bookings and 100 containers are scheduled. We let the cost on the non-horizontal truck arcs be equal to the travel time of the arc. This models a company that owns barges and has to pay additional cost when trucking containers. We make sure that it is possible to truck a booking to its destination, so its deadline should be (at least one time stamp) later than its release date. Besides the terminal locations, there are two customer locations in the model. The customer locations are not reachable by barge. So in the time-space network none of the barge arcs are incident to customer locations (that are not accessible by barge). Trains or other (types of) vehicles are initially not in the model, but could easily be added. We solved our ILPs with IBM's CPLEX solver (IBM, 2017).

In all the experiments reduction I is used, because it is not beneficial to add variables that cannot be non-zero in any feasible solution. The variables that are removed by reduction I and those that are removed by reduction E can never be used even if they are included in the model. The solver CPLEX removes many of those variables already automatically in the pre-processing phase. Therefore, these two reductions may influence the time for the pre-processing more than the time required for the actual solving. Though, an experiment in the past did show that reduction I also reduced the computation time of the solving phase. We will turn the reductions A-G on and off to see how they influence the computation time. In Table 1 the results are shown. Computation times are the total computation time for solving the problem on a DELL E7240 laptop with an Intel(R) Core(TM) i5-4310U CPU 2.00 GHz 2.60 GHz processor. The laptop is operational on a 64-bit operating system.

Table 1: Numerical results of reductions

Reduction	Active	Parameter	Comp. Time	Solution
A	No	K=25	7.12s	2600 (opt.)
A	Yes	K=25 → 20	5.86s	2600 (opt.)
A	No	K=50	67.45s	3760 (opt.)
A	Yes	K=50 → 39	61.16s	3760 (opt.)
B	No		61.16s	3760 (opt.)
B	Yes		43.35s	3760 (opt.)
C	No	$\|W\|$=6	1667.61s	3760 (opt.)
C	Yes	$\|W\|$=5	628.58s	3760 (opt.)
C	Yes	$\|W\|$=4	183.51s	3760 (opt.)
C	Yes	$\|W\|$=3	61.16s	3760 (opt.)
D	No		117.61s	3760 (opt.)
D	Yes	Sink Incoming	61.16s	3760 (opt.)
D	Yes	Sink In/Out	64.58s	3760 (opt.)
D	Yes	Complete	58.50s	3760 (opt.)
F	No		129.98s	3760 (opt.)
F	Yes		61.16s	3760 (opt.)

G	No	> 300.00s	-
G	Yes	61.16s	3760 (opt.)

Reduction A is used to reduce the initial number of commodities $|K| = 50$ to $|K| = 39$. For this reduction the computation time of the ILP with this reduction and without are compared and we also split some commodity into sub commodities to investigate the effect of taking a larger set $|K|$ on the computation time. Our set of vehicle types W = {truck, typeAbarge, typeBbarge} is obtained by applying reduction C. If we only merge some of those barges of the same type or none of them, then we are using reduction C partially or not at all. In those cases the set of vehicle (types) W is larger as we see in the table in the comparisons for reduction C. Reduction D is also implemented completely and partially to get some better insight in the effect of the reduction.

5 Conclusion

Reduction A seems to help a little, but clearly is most effective if there are many bookings with the same/similar sinks. The reduction may be less effective than expected, because adding multiple sources for a commodity may make the problem more difficult to solve. From theory it is expected that reduction A is beneficial, if it is the number of commodities that blows up the computation time and many bookings have the same sink location. Reduction B surprisingly leads to better results, though it increases the number of variables because it was implemented together with reduction I. Reduction C is a very powerful tool. Reduction C does require a company to have many identical vehicles, though. Reduction D halves the computation time for our instance. Reduction F is effective, though we do need to do pre-calculations to use it. Reduction G is very effective. We do need however to know the structure of the waterways to use it. Reduction H is expected to do well just like other location reductions. It is however possible that good solutions will be removed by using this reduction. Reduction J may be the most powerful reduction of them all. If the time windows at which locations can be visited are small, then many variables will disappear from the problem. The effectiveness of this reduction depends on the number of time stamps at which terminals are accessible. If there are few of those moments, than it will probably drastically reduce the computation time.

Acknowledgements

This work has been carried out within the project ‘Complexity Methods for Predictive Synchromodality’ (Comet-PS), supported and funded by NWO (the Netherlands Organisation for Scientific Research), TKI-Dinalog (Top Consortium Knowledge and Innovation) and the Early Research Program ‘Grip on Complexity’ of TNO (The Netherlands Organisation for Applied Scientific Research).

References

- Andersen, J., T.G. Crainic, M. Christiansen (2009): Service network design with asset management: Formulations and comparative analyses. Transportation Research Part C: Emerging Technologies 17, 197–207.
- Babonneau, F., O. du Merle, J. Vial (2006): Solving large scale linear multicommodity flow problems with an active set strategy and proximal-ACCPM. Operations Research 54, 184–197.

- Bai, R., G. Kendall, R. Qu, J. Atkin (2012): Tabu assisted guided local search approaches for freight service network design. Information Sciences 189, 266–281.
- Balakrishnan, A., T. Magnanti, R. Wong (1989): A dual-ascent procedure for large-scale uncapacitated network design. Operations Research 37, 716–740.
- Chouman, M., T.G. Crainic (2012): MIP-based Matheuristic for Service Network Design with Design-balanced Requirements. CIRRELT.
- Crainic, T.G. (2000): Service network design in freight transportation. European Journal of Operational Research 122, 272–288.
- Crainic, T.G., A. Frangioni, B. Gendron (2001): Bundle-based relaxation methods for multicommodity capacitated fixed charge network design. Discrete Applied Mathematics 112, 73–99.
- Crainic, T.G., M. Gendreau, J. Farvolden (2000): A simplex-based tabu search method for capacitated network design. INFORMS Journal on Computing 12, 223–236.
- De Juncker, M. A., D. Huizing, M.O. del Vecchyo, F. Phillipson, A. Sangers (2017): Framework of synchromodal transportation problems. International Conference on Computational Logistics, 383-403.
- Ghamlouche, I., T.G. Crainic, M. Gendreau (2004): Path relinking, cycle-based neighbourhoods and capacitated multicommodity network design. Annals of Operations research 131, 109–133.
- Hewitt, M., G.L. Nemhauser, M.W.P. Savelsbergh (2010): Combining exact and heuristic approaches for the capacitated fixed charge network flow problem. Journal on Computing 22, 314–325.
- Holmberg, K., J. Hellstrand (1998): Solving the uncapacitated network design problem by a lagrangean heuristic and branch-and-bound. Operations research 46, 247–259.
- Holmberg, K., D. Yuan (2000): A Lagrangian heuristic based branch-and-bound approach for the capacitated network design problem. Operations Research 48, 461–481.
- Huizing, D. (2017): General methods for synchromodal planning of freight containers and transports. Master's thesis. TU Delft.
- IBM (2017): Cplex optimizer. URL: https://www.ibm.com/analytics/data-science/prescriptive-analytics/ cplex-optimizer.
- Janic, M. (2007): Modelling the full costs of an intermodal and road freight transport network. Transportation Research Part D: Transport and Environment 12, 33–44.
- Joborn, M., T.G. Crainic, M. Gendreau, K. Holmberg, J.T. Lundgren (2004): Economies of scale in empty freight car distribution in scheduled railways. Transportation Science 38, 121–134.
- Li, X., K. Wei, Y. Aneja, P. Tian (2017): Design-balanced capacitated multicommodity network design with heterogeneous assets. Omega 67, 145–159.
- Magnanti, T.L., R.T. Wong (1984): Network design and transportation planning: Models and algorithms. Transportation Science 18, 1–55.
- Page, L., S. Pichai (2018): Google maps. URL: https://www.google.nl/maps.
- Pedersen, M., T.G. Crainic, O. Madsen (2009): Models and tabu search metaheuristics for service network design with asset-balance requirements. Transportation Science 43, 158–177.
- Rodríguez-Martín, I., J.J. Salazar-Gonzalez (2010): A local branching heuristic for the capacitated fixed-charge network design problem.´ Computers and Operations Research 37, 575–581.

- Sharypova, K. (2014): Optimization of Hinterland Intermodal Container Transportation. Ph.D. thesis. Eindhoven University of Technology.
- Vu, D.M., T.G. Crainic, M. Toulouse (2012): A three-stage matheuristic for the capacitated multi-commodity fixed-cost network design with design-balance constraints. CIRRELT .
- Vu, D.M., T.G. Crainic, M. Toulouse, M. Hewitt (2014): Service network design with resource constraints. Transportation Science 50, 1380–1393.