Les cahiers Leibniz

A Decomposition Approach solving a Large Scale Service Network Design with Asset Management problem

Nicolas Teypaz, Susann Schrenk, Van-Dat Cung

Laboratoire G-SCOP
46 av. Félix Viallet, 38000 GRENOBLE, France
ISSN : 1298-020X

n° 170

Juillet 2008

Site internet : http://www.g-scop.inpg.fr/CahiersLeibniz/

A Decomposition Approach solving a Large Scale Service Network Design with Asset Management problem *

Nicolas Teypaz, † Susann Schrenk ‡and Van-Dat Cung
G-SCOP, Laboratoire des Sciences pour la Conception, l'Optimisation et la Production, INPG-UJF-CNRS,
46, avenue Felix Viallet, 38031 Grenoble, France.
{nicolas.teypaz, susann.schrenk, van-dat.cung}@g-scop.inpg.fr

Abstract

In this paper, we address a complete freight transportation problem. The objective function consists of maximizing the carrier profit. It is not necessary to satisfy all demands but profitable demands should be satisfied globally or partially. The carrier needs to ensure a return on his vehicle investment: his vehicles must be neither under- nor over-used. To tackle this industrial problem, we present a model based on a formulation of the Service Network Design with Asset Management problem. Due to the difficulty of finding optimal solutions to this problem, we propose a decomposition of the problem into three main steps: construction of the network, filling vehicles with commodities, construction of the vehicle plannings. The resolution of these steps involves heuristic schemes, Mixed Integer Programming and Constraint Programming. We consider different methods for solving these steps and whether to allow transhipment. To evaluate the model and the solution algorithms, we produced instances based on a study of real data. The results show that the methods are robust without transhipment and that transhipment can improve profit.

Keywords: Freight transportation, Service Network Design, Asset Management

1 Introduction

1.1 Freight transportation

Historically, the factories themselves were in charge of the distribution of their products to their clients (wholesalers and retailers). However, with increasing competition and rationalization of resources, the companies' priorities are product design and production, whereas distribution is increasingly subcontracted to logistics and transportation businesses. Carriers will transport a certain amount of freight, depending on the demands, between the production sites and the customers. Transportation costs depend mainly on distances. In order to minimize them, the carrier has to optimize utilization of its transportation media (i.e. to maximize the filling of its vehicles). He may consolidate some commodities (transhipment) so as to concentrate the flow of goods and to reduce the covered distances as a consequence.

The problem arises from the following industrial application, related to regular freight transport plannings. In our study, we deal with the determination of the transportation plan for regular operations. The transportation plan is constructed for a given planning horizon (often one day, one week or one month) from a set of selected transportation services. The transportation plan is replicated for each planning horizon. Those tactical planning problems are generally modeled as a Service Network Design Problem [Wieberneit, 2008]. A carrier is contracted for:

*This research has been supported by the following grants and contracts: BQR INPG "Optimisation du transport de fret par l'utilisation de plateformes logistiques"; Cluster de Recherche GOSPI, Gestion et Organisation des Systèmes de Production et de l'Innovation, Région Rhônes-Alpes.

†BQR INPG "Optimisation du transport de fret par l'utilisation de plateformes logistiques "

‡Cluster de Recherche GOSPI (Gestion et Organisation des Systèmes de Production et de l'Innovation), Région Rhônes-Alpes

"deliveries from a some number of factories to a some number of clients" which constitutes the transportation demand (i.e. commodity). We suppose that the origins and destinations of all demands are terminals : harbors, rail stations, airports or warehouses. We suppose that transportation demands are periodic : the demands are repeated for every planning horizon (demands are made for raw materials, semi-finished products and customer products). In our problem, the demand between two terminals is an estimation of transportation demand. As a consequence, we allow the carrier to reject some demands completely or partially. Thus, the carrier is able to increase profits. The resolution of this problem provides a feedback on the transportation demands. This feedback allows us to build up regular services and to adapt the fleet size in order to maximize the carrier profit.

Every time the carrier takes charge of the transport of a freight unit, he is paid by the shipper, but he has to pay the transportation costs (fuel, vehicles, crew...). The questions the logistic business (carrier) has to answer are : Which transportation demands satisfy every planning horizon ? Are they completely or partially satisfied ? How should the chosen freight be carried ? Which transportation services are the best ones ? How to plan all the rides ?

The freight transportation problem we deal with has additional characteristics. The first concerns transportation costs, they depend not only on the distance covered but may also depend on quantity on board. The fixed cost is the cost to organize one transportation service (the crew cost, the handling cost and the fuel cost). The variable cost depends on quantity on board, this corresponds to the additional fuel. Note that we do not have transhipment costs in this problem.

The other characteristics of our problem are two time-related constraints. Periodicity of demand implies that if a commodity is loaded during a planning horizon, it has to be delivered in the same planning horizon. Furthermore, as the demand is periodic, we have to satisfy exactly the same demands every planning horizon, in other words, provide exactly the same transport services. To guarantee this, vehicles should be spread out in an analogous way at the beginning and at the end of a planning horizon. Thus the transport can continue in the next planning horizon in exactly the same way. It is not necessarily the same vehicle which has to be in the same location, vehicles may rotate. This often happens in transportation for sea [Chou et al., 2003], rail, and air [Yan et al., 2005] media. This rarely occurs for road transportation, the same vehicle can run several routes starting at the same terminal in one period. To ensure vehicle rotation, it may be necessary to make provisions for repositioning of some vehicles [Crainic, 2000].

A second time constraint is that carriers want to use all their vehicles evenly. At the end of a given period (month or year) no vehicle should be neither under- nor over-used. We enforce time utilization of any vehicle to be in a certain interval. This constraint means that the length of any vehicle route is between two bounds. In vehicle routing problems literature, this constraint is referred to as the time- or distance-constraint [Laporte, 1992]. We will take this additional constraint into account.

In our resolution, we suppose that the carrier has an infinite fleet, nevertheless the fair use of vehicles explained above will limit the number of used vehicles. Thus, our solution may help the carrier in the management of his fleet : Which vehicles should be sold ? Which new vehicles should be bought ? And if his present fleet is appropriate, the provided solution will increase profits.

The paper is organized as follows. Section 1.2 introduces the Service Network Design problem and its extensions (Section 1.2). This problem is similar to our freight transportation problem except two additional particularities : satisfied demands are rewarded and each vehicle used must operate a transportation service times between two bounds during the planning horizon. The contributions of our study are displayed in Section 1.3. We present a model for this NP-hard problem in Section 2 and propose a resolution scheme which allows us to determine the transportation services, choose which demands to satisfy and schedule the transportation services. Section 3 is dedicated to the decomposition of the problem into three main steps, which are detailed in Section 4. To conclude, Sections 5 and 6 present computational results of our method, certain limits and some avenues for further study.

1.2 Service Network Design

Service Network Design problems (SDNP) are essential in the construction of a transportation network [Crainic, 2002]. Their formulations are associated with the long-term evolution of transportation infrastructures and services. Service Network Design is often used to solve freight transportation problems with consolidation : by rail [Barnhart et al., 2000], by air [Barnhart et al., 2002, Jaillet et al., 1996] and also by road. The objective of Service Network Design problems is to select links in a network, along with capacities, in order to satisfy transportation demands at the lowest possible system cost. A transportation demand is composed of : the nature of the goods, the size of the load to be transported, its origin and its destination, seen as terminals. The system cost includes fixed and variable costs. Fixed cost indicates that as soon as a link is selected, one has to pay for the transport on this link. Most of the time, variable cost depends on quantity on board. Flows of commodities are in the formulation, and the transportation route of a commodity uses the open services. Thus, when the services are assigned to vehicles, commodities may change vehicle. In other words, transhipment is taken into account.

The Service Network Design has been enlarged with the management of the vehicles (or assets) needed to operate the transportation services [Andersen et al., 2007c]. Asset management is important because of the high acquisition cost of vehicles. In the literature, the design balance constraints are introduced in order to ensure that there is an equal number of vehicles entering and leaving each terminal (node) in the network [Pedersen et al., 2007, Andersen et al., 2007a]. Design balance constraints have been modeled for an application of express freight service by air [Barnhart and Schneur, 1996, Barnhart et al., 2002], by road [Smilowitz et al., 2003] and for an application of passengers transportation service by ferry [Lai and Lo, 2004].

The formulation of the Service Network Design can be extended to a time-space network [Andersen et al., 2007c]. The planning horizon is divided into a set of time periods, and the terminals in the static network are duplicated in each time period. Service arcs represent movements in both time and space. In addition, waiting arcs between consecutive time realizations of physical terminals represent the possibility that vehicle and flow wait at a terminal (see the Figure 1). This problem can be considered as the design of a scheduled service network for a transportation system where each vehicle operates several consecutive services during the planning horizon.

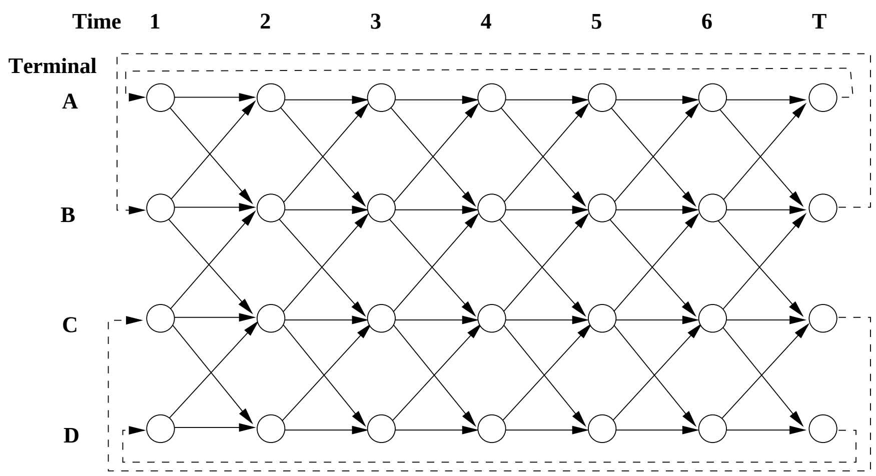

Figure 1: Time-space network.

As can be seen, time introduces a significant increase in the number of variables. Indeed, to take time into account we need to index any variable with a time parameter (after discretization of time) [Hahgani and Oh, 1996]. Discretization is needed to guarantee that the delivery is done in the same planning horizon as the pick-up as shown in Section 4.2.2. It allows us to exhibit the planning of each vehicle for the planning horizon. In Section 2, we will present this global formulation which, while perfectly adapted to our problem, also has a large number of variables and constraints as explained above.

1.3 Contributions

The specificity of this study is that it deals with the whole freight transportation problem. Thus we consider the underlying network design problem as well as the scheduling of the different transportation services chosen. Furthermore, we can choose the most profitable demands and reject others partially or globally. This possibility is rarely studied in transportation problems.

Thus, we deal with both, the specifics of multicommodity network design problems and the specifics of vehicle routing problems. In Service Network Design problems, we define the routes for each demand : at each visited node, transhipment operations are a possibility. The routes of vehicles which carry all satisfied demands correspond to vehicle routing problems. Moreover each route must conform to the time-constraints.

Concerning applications, our approach can provide a global transportation solution and can solve realistically-sized problems in freight transportation (Section 3).

Concerning performance, preliminary numerical tests have shown that the global resolution of the problem cannot reach a solution (optimal or not) in an acceptable times. Our smallest instance studied contains almost 140 demands and 40 terminals, with the time horizon divided into 480 time periods. In [Andersen et al., 2007b], the greatest instance used to evaluate the performance of different formulations contains 200 demands and 25 time periods. As a consequence, our main proposal is to decompose the problem into three steps as follows :

- Construction of the network (Section 4.1) ;
- Choice of the transported commodities (Section 4.2) ;
- Construction of the vehicle plannings (routes and scheduling, Section 4.3).

Moreover, this decomposition allows to easily integrate improvements of one of the step and to improve the resolution method. Resolution involves many different tools of Operations Research : graph algorithms, Linear and Mixed Integer Programming, column and row generation and Constraint Programming. In addition to this decomposition, we also studied how these steps can be performed in succession. The order in which we solve these three steps decides whether or not transhipment can be used.

2 Model of the problem

This section presents a model for the freight transportation problem studied in this paper as a single objective linear programming model on a time-space network. This model is based on the formulation of the Service Network Design with Asset Management [Andersen et al., 2007c]. We have completed this formulation by taking into account the particularities of our problem :

- the carrier can reject some demands completely or partially,
- no vehicle must be neither under- nor over-used (time-constraints).

The freight transportation problem can be considered as NP-hard if we assume that the number of transportation services is bounded by a polynomial expression $f(n)$ where n is the number of terminals. Otherwise the solutions of our problem cannot be checked efficiently in polynomial-time because in order to check the time-constraints, we must run each vehicle schedule.

2.1 Notation

The periodic transportation demand k is as follows : for every planning horizon (week), transport the quantity w^k unities of commodities, from origin $O(k)$ to destination $D(k)$. Each time the carrier satisfies one unit of demand k (one tonne, one container, one barrel...), he earns quantity P^k of money. Let K be the set of all demands. Origin and destination are terminals on which carrier has access. The distance between two terminals is $d(i, j)$. We denote the set of terminals by V ($|V| = n$), they form the vertices of a directed graph $G = (V, E)$ where E is the set of arcs which make the graph complete ($|E| = m$). Connections are done by one of several transportation means : road, rail, sea or air.

To transport the freight we have a set L of different vehicles which are classified by their types. We can assume that the number of vehicles is large. We denote by L_{type} the set of

vehicles of types. We assume that the vehicles are spread out as wished at the beginning of the planning horizon. Each vehicle $l \in L$ is able to transport a maximum of C^l (capacity of vehicle of type l) units and needs time $t^l(i,j)$ to go from origin i to destination j. Once arrived at j, the vehicle has to stay there for at least τ time units before continuing its trip. This is for the vehicle to be refueled and for cargo to be loaded and unloaded. This break time may include a buffer to reduce lateness.
If a vehicle l carries w units (from several demands w^k) directly from terminal i to terminal j, it costs the fixed cost $F_f^l(i,j)$ plus the variable cost $F_v^l(w,i,j)$. For any vehicle l, variable costs depend on quantity on board and on covered distances (the more the vehicle carries and the farther it goes, the more fuel is needed and the more it costs to the carrier). The variable cost is assumed to be linear with the weight of carried freight: we denote $U^l(i,j)$ the variable cost per unit of commodity transported between i and j: thus $F_v^l(w,i,j) = U^l(i,j) \cdot w$.

Over a planning period, if a vehicle l is used, it has to fulfill a specific quota of hours defined by the minimum $Quota^l_{min}$ and the maximum $Quota^l_{max}$ quota of hours. These quotas depend on the vehicle type.

2.2 Formulation as a Mixed Integer Program

The formulation as a Mixed Integer Program (MIP) is adapted from Service Network Design problems. As stated in the previous section, we need to index all variables with a time parameter. We denote by $\mathcal{T}$ the set of discretized time over the planning horizon. The variables we use are as follows:

- At time t the fraction x^{kl}_{ijt} of commodity k is charged in origin i for destination j by vehicles of type l. Variables x^{kl}_{ijt} are between 0 and 1 and represent flows of commodities.
- The boolean variables y^l_{ijt} indicate whether the vehicle l is going from origin i to destination j at time t. Variables y^l_{ijt} choose the service for the vehicle l (i.e. define the network).
- The boolean variables δ_l indicate whether or not the vehicle l is used.
- z^k represents the fraction of demand k which is carried during one planning horizon, that is the fraction of demand k which is satisfied.

The carrier wants to maximize his profit (which is the revenue minus the fixed and variable costs) by respecting all constraints as detailed below.

Maximize:
Profit= revenue - fixed and variable costs

$$\sum_{k \in K} P^k \cdot w^k \cdot z^k - \sum_{(ij)\in E} \sum_{l \in L} \sum_{t \in \mathcal{T}} \left(F_f^l(i,j) \cdot y^l_{ijt} + \sum_{k \in K} U^l(i,j) \cdot w^k \cdot x^{kl}_{ijt} \right) \tag{1}$$

subject to:
Multicommodity flow constraints:

$$\sum_{j \in \delta^+(i)} \sum_{l \in L} x^{kl}_{ijt} - \sum_{j \in \delta^-(i)} \sum_{l \in L} x^{kl}_{ji(t-t^l(j,i)-\tau)} = 0 \quad \forall i \in V, i \neq O(k), i \neq D(k), \forall k \in K, t \in \mathcal{T} \tag{2}$$

$$\sum_{t \in \mathcal{T}} \sum_{j \in \delta^+(i)} \sum_{l \in L} x^{kl}_{ijt} - \sum_{t \in \mathcal{T}} \sum_{j \in \delta^-(i)} \sum_{l \in L} x^{kl}_{jit} = \left\{ \begin{array}{lll} z^k & \text{if} & i = O(k) \\ -z^k & \text{if} & i = D(k) \end{array} \right. \quad \forall i \in V, k \in K \tag{3}$$

Vehicle route constraints:

$$\sum_{(ij)\in E} \sum_{t' \in \mathcal{T} : t' \leq t < t' + t^l(j,i) + \tau} y^l_{ijt'} + y^l_{iit} - \delta_l = 0 \qquad \forall t \in \mathcal{T}, l \in L \tag{4}$$

$$\sum_{j \in \delta^+(i)} y^l_{ijt} + y^l_{iit} - \sum_{j \in \delta^-(i)} y^l_{ji(t-t^l(j,i)-\tau)} = 0 \; \forall t \in \mathcal{T}, t \neq 1, t \neq |\mathcal{T}|, i \in V, l \in L \tag{5}$$

$$Quota^l_{min} \delta_l \leq \sum_{t \in P} \sum_{(ij)\in E} t^l(i,j) \cdot y^l_{ijt} \leq Quota^l_{max} \delta_l \quad \forall l \in L \tag{6}$$

Design-balance constraints on the vehicles types :

$$\sum_{l\in L_{type}}\sum_{t\in\mathcal{T}}\sum_{j\in\delta^+(i)} y^l_{ijt} - \sum_{l\in L_{type}}\sum_{t\in\mathcal{T}}\sum_{j\in\delta^-(i)} y^l_{jit} = 0 \qquad \forall i\in V,\, type \tag{7}$$

Bundle constraints :

$$\sum_{k\in K} w^k\cdot x^{kl}_{ijt} \le \sum_{l\in L} C^l\cdot y^l_{ijt} \qquad \forall (ij)\in E,\, l\in L,\, t\in\mathcal{T} \tag{8}$$

Variables :

$$y^l_{ijt}\in\{0\,;1\} \qquad \forall (ij)\in E,\, l\in L,\, t\in\mathcal{T} \tag{9}$$

$$x^{kl}_{ijt}\in[0..1] \qquad \forall (ij)\in E,\, l\in L,\, k\in K,\, t\in\mathcal{T} \tag{10}$$

$$z^k\in[0..1] \qquad \forall k\in K \tag{11}$$

$$\delta_l\in\{0\,;1\} \qquad \forall l\in L \tag{12}$$

Constraints (2) and (3) are flow constraints : (2) requires that demand k arrived at terminal i ($i\neq O(k)$ and $i\neq D(k)$) from j ($t^l(j,i)$ is transportation time between j and i) will not remain at i in the next time period. A break of at least time τ is necessary before transportation can continue. This allows some transhipment , so long as it is synchronous as storage is not allowed at terminal i. To model storage, the variables x_{iikt} would be needed, in which case asynchronous transhipment would also be possible. In our study, we consider the asynchronous transhipment but with zero storage costs. The expected fraction z^k of demand k starts from $O(k)$ and arrives at destination $D(k)$ with respect to transportation time (3). Allocation of vehicles according to their capacities is the consequence of constraints (8).

Constraints (4) state that for each time period, if a vehicle l is used, it should be engaged in one activity only : the vehicle does one transportation service between two terminals ($y^l_{ijt'}=1$) or the vehicle waits in the last terminal ($y^l_{iit}=1$). During one planning horizon, we must satisfy the design-balance constraints (5) for each vehicle used. We notice that the design-balance constraints for each vehicle are not checked for $t=1$ and $t=|\mathcal{T}|$ because the vehicle rotation is allowed (see Figure 2). Constraints (6) ensure that the transport time is between $Quota^l_{min}$ and $Quota^l_{max}$ for each vehicle l used. We will refer to these constraints (6) as the time-constraints. These constraints permit us to define the route of each vehicle.

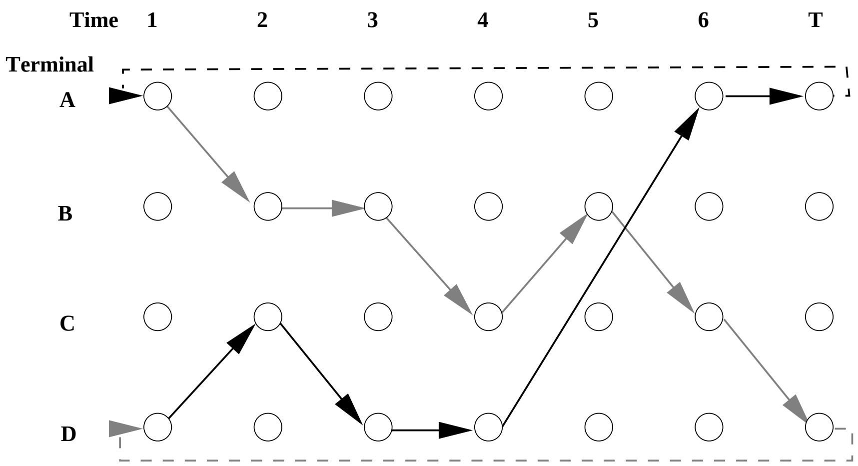

Figure 2: Illustration of the vehicle rotation with two vehicles of the same type.

The vehicles of the same type have to do cycles[1] such that their transportation plannings can be repeated planning horizon after planning horizon without the need for an expensive repositioning of a vehicle by doing a trip with empty load (Section 1.1). We will refer to these constraints (7) as the design-balance constraints. The Figure 2 highlights the vehicle rotation permitted by these constraints.

[1]For the sake of simplicity, all cycles considered in this paper are directed.

2.3 An upper bound for our problem

As this formulation has too many variables and constraints, mainly due to time index, we cannot expect any response for real transportation instances by solving this MIP directly. Nevertheless, the Service Network Design with asset balance requirement formulation, without taking time into account, allows us to get an upper bound for global profit. The smallest instance studied with this timeless model requires approximately 225 000 variables including 1600 integer variables and approximately 7200 constraints.

Its resolution provides only commodities to carry, vehicle types and related networks and does not produce the transportation planning. There is no guarantee that a complete planning of vehicles, which fulfills all these choices, exists. However, it does choose which commodities and transportation network would give the largest profit. Thus the profit of any feasible solution cannot be greater than the profit of the found solution, even though it may be infeasible. We calculated the upper bound by solving an aggregate form of Service Network Design formulation without time index with ILOG Cplex. The aggregated formulation reduces the smallest instance above to approximately 68 000 variables including 1600 integer variables and approximately 3200 constraints.

The computed upper bound is used in Section 5 to measure the quality of our approach based on the decomposition of the problem as presented in next section.

3 Decomposition of the problem

Our resolution method is based on a decomposition of the problem into three main parts. An overview of our approach follows which will be detailed further in subsequent sections. This method not only allows us to take into account all the constraints of our problem but also to display solutions in progressive stages. Indeed, the solution contains two sets : the vehicle plannings and the satisfied demand routes. Our approach seems more appropriate than meta heuristics where the representation of solutions is not trivial.

1. **Choose which transportation services will be open (i.e. construct the service network)**. In this step, the frequencies of transportation services y_{ij} are established for each vehicle type. To be able to repeat the transportation planning from planning horizon to planning horizon, each vehicle type should perform one or more cycles. To ensure this, in the network constructed for each vehicle type, the number of in-coming services must be equal to the number of out-going services for each terminal.

   We proceed first by constructing the network of the most profitable vehicle type (the cheapest per transport unit per distance unit), then we iterate the network construction for the remaining vehicle types. The constructed network is set once and for all, thus the fixed cost is completely determined after this step.

   <u>Remark :</u> the obtained network has Eulerian cycle properties, thus the design-balance constraints are respected, but Eulerian cycles are constructed only in step 3.

2. **Fill vehicles with commodities (i.e. choose which freight to carry)**. From the transportation services network, we maximize the revenue, we determine the satisfied fraction of each demand z^k. Each transportation service a allows us to carry a volume x_a^k for each commodity k.

   We have to take care that each commodity fraction is transported by only one vehicle and that vehicle's capacities are respected. Transporting a certain amount of freight will bring in a certain amount of revenue. Profit is maximized in this step. The choice of which freight to carry is solved as a maximum flow problem on an appropriated network.

3. **Construct the vehicles plannings (i.e. schedule the different routes)**. We assign each transportation service y_{ij} to one vehicle l and one departure time t. This means that the set of consecutive transport services y_{ijt}^l is decided for each vehicle l.

   In this step we consider the vehicles individually. We construct the vehicle routes and the corresponding schedules, which we call the vehicle planning. In order to facilitate planning, we start by construction of the Eulerian cycle of each fleet network (and alternatively the

Eulerian cycles of the connected components). On a given cycle, time quotas are easily taken into account. We use Constraint Programming to solve this step.

The first step of the resolution algorithm is to design the network. Once this has been done, we may choose in which order we do the two remaining steps according to whether or not transhipment is considered. In any case they are not independent. In the next section, we consider both possibilities and detail the different steps and their characteristics.

4 Resolution algorithms

The two possible orders of resolution steps mentioned above are described in Figures 3 and 4. In this section, we detail the three steps for both methods and consider some points which must not be overlooked.

The first algorithm performs the steps in order 1-3-2 : after designing the network it gives independent transportation plannings respecting time quotas and finally it fills the plannings with freight. There is no interaction between the different vehicle types, no transhipment is considered.

In the second method, we inverse the order of last two steps. Thus the second decision is the filling of the network with commodities. This has to be done carefully, we have to be sure that the chosen commodities can be carried and delivered in the same planning horizon. This will be detailed in section 4.2.2. Interaction between vehicles exists : some commodities may be transferred from one vehicle to another. This implies that transportation plannings must allow for transfer operations. Thus, transhipment complicates construction of vehicle plannings, and creates some precedences between transportation services. Of course, time quotas still have to be respected.

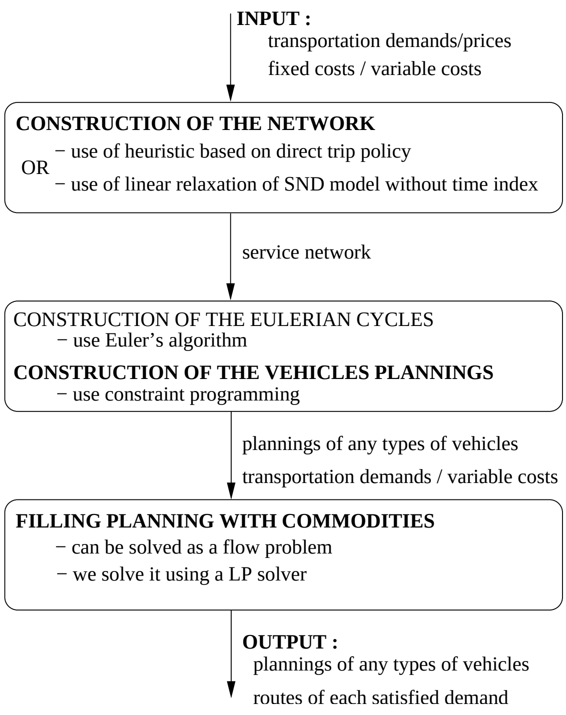

Figure 3: Algorithm without transhipment.

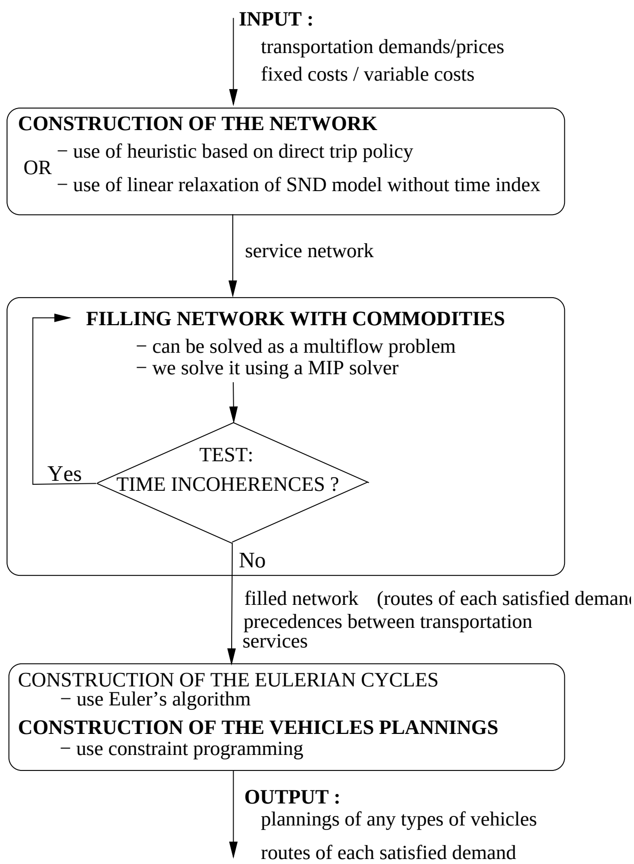

Figure 4: Algorithm with transhipment.

4.1 Construction of the service network

We propose two different methods for the design of the service network. The first is based on an heuristic which searches for profitable cycles. This method provides us with a value that we consider to be the lower bound for our problem. From the results of the first step, we can arrive at a solution where each satisfied demand is carried directly from its origin to its destination. In the second we solve a linear relaxation of the Service Network Design formulation without time index as for the upper bound (see section 2.3).

4.1.1 Direct trip policy - heuristic for constructing the network

We construct the service networks successively for the different vehicle types. Let l' be the first vehicle type for which we construct the service network. In practice we start by constructing the service network of the most profitable type (that is the vehicle, which, once filled, is the least expensive per transport unit per distance unit). Then, once the service network is constructed for the vehicles of type l', we remove the commodities it deals with and apply the same procedure to construct the networks for the other vehicle types.

We suppose that any commodity is transported directly from its origin to its destination by a vehicle of type l'. As there is no stop-over for any commodity, we need $\lceil \frac{w^k}{C^{l'}} \rceil$ vehicles of type l' to transport the total demand w^k of product k between its origin $O(k)$ an its destination $D(k)$. Each of these virtual transport operations of a fraction of demand k induces a transportation cost (fixed + variable) and brings in the revenue of the satisfied fraction of the demand k. Thus, each transport operation corresponds to a profit (which is positive or negative).

We consider a graph in which the vertices are terminals, and the description above explains how we construct weighted arcs. We complete this graph with arcs corresponding to empty transports (their cost is only the fixed transportation cost) such that there are exactly the same number ($\lceil \frac{max_k w^k}{C^{l'}} \rceil$) of arcs between each pairs of vertices. These extra arcs will allow repositioning of vehicles with empty load.

In this graph, some arcs are weighted by a positive value (which corresponds to profit) and others by a negative value (loss). As transportation plannings have to be copied from one planning horizon to the other and due to vehicle repositioning constraints, the chosen arcs (which will compose the network) have to form one or more cycles. Moreover, to maximize profit we choose cycles with global positive weight. Now, to solve this graph problem optimally, we modeled it as a maximum cost flow problem and solved it using the *minimum mean cycle canceling* algorithm [Ahuja et al., 1989]. This algorithm is polynomial $\mathcal{O}\left(n^2 m^3 \log(n)\right)$ in size of the constructed graph (n vertices and m arcs), but the number of arcs of this graph (depending on $\lceil \frac{max_k w^k}{C^l} \rceil$) is pseudo-polynomial in size of the freight transportation problem. The obtained optimal value after this step is considered as the lower bound.

4.1.2 Construction of the service network - Linear relaxation

In order to measure the quality of our heuristic, we compare our result with the network constructed by a linear relaxation of the aggregated Service Network Design formulation with the design-balance constraints (without time index), as for the upper bound (section 2.3).

We did not explore this method in depth. To obtain the integer solution, after each resolution step using ILOG Cplex we round out variables with a fractional part less than 0.25 or greater than 0.75. If no variable can be rounded, we round the variable which is nearest to an integer. We will present the results corresponding to both constructions of the network in Section 5.

4.2 Choice of the freight to carry

Once the service network is constructed by the direct trip policy or linear relaxation, we have to choose which commodities will be carried. We have two possibilities: either we allow transhipment or not. We look at both to measure how transhipment can increase profit.

4.2.1 Without transhipment

Without transhipment the choice of freight is done at the last step and corresponds to optimally loading the planned vehicles (see Figure 3). At the first step, a profitable network is constructed considering only direct trips. However, more profits could be achieved by considering indirect trips and possibilities of loading and unloading at each terminal. This is done by solving a maximum cost flow problem. From the planning of one vehicle, we build one particular instance of the maximum cost flow problem (see Figure 5) in order to fill this vehicle with demands. Each arc links the origin to the destination of one demand. The capacity of the arc corresponds to the demand size and the cost to the unit profit of the demand (the revenue minus the variable costs). We also consider empty loads represented by the dashed arcs $(\infty, 0)$. We add two vertices: one source s and one hole t

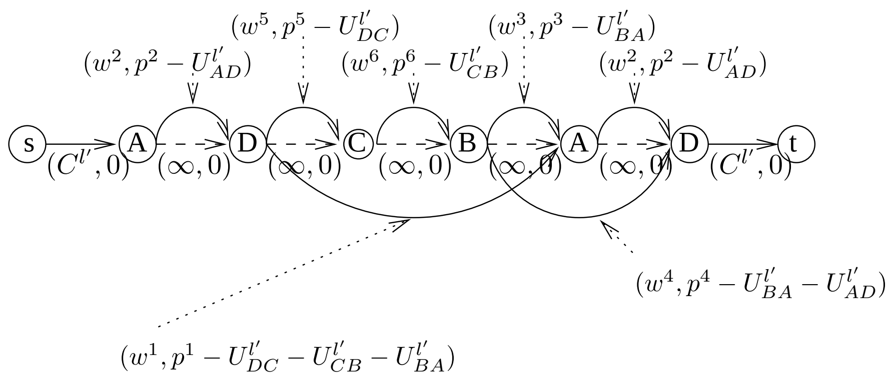

Figure 5: Maximum flow problem for one vehicle.

As, by construction, no interaction exists between the different plannings, we can fill them at the same time by solving a maximum flow problem: we append the different flow problems and introduce a super-source and a super-hole. This maximum cost flow problem contains some specific capacity constraints in order not to exceed certain demands. For example, these specific constraints must be set for the demand from A to D (see Figure 5). The flows over the two arcs AD must not globally exceed the capacity w^2.

Thus this step is done in polynomial time in size of the inputs, but as mentioned before, the size of this service network is pseudo-polynomial according to the inputs of the freight transportation problem.

4.2.2 With transhipment, time incoherences may occur

To make transhipment possible, we inverse the order of the last two steps (see Figure 4). Once the service network is designed we determine which commodities to transport on each arc. This is done by solving a multicommodity flow problem in an oriented multiple edged graph, where the objective is to maximize profit and each edge of the graph corresponds to one transportation service. This problem is polynomial in size of inputs, it can be solved either by using Linear Programming (LP), or by using approximation algorithm [Garg and Konemann, 1998, Fleischer, 2000].

However, when choosing which freight to carry, no time constraints are taken in account, and some time incoherences may occur which would interfere with the construction of plannings in the last step. They are described in the next paragraph in an example. These have to be identified and, as a result, some commodity choices may have to be revised. This is managed with constraint and variable generations. This is examined in further detail and illustrated in an example in the following paragraph.

Description of time incoherences Time is not considered neither when we construct the service networks nor when we fill service networks with commodities. Thus, let A, B, C and D denote four terminals, the situation represented in Figure 6 may occur.

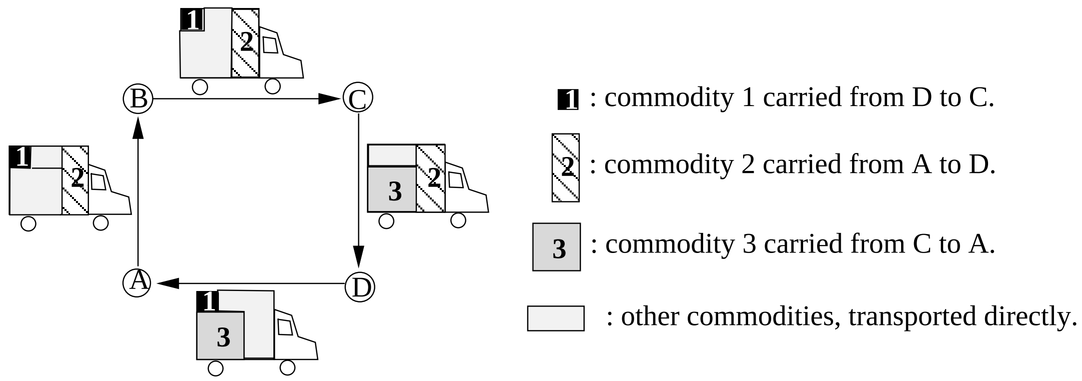

Figure 6: Time incoherences.

The same vehicle has to transport the different represented commodities from their origin to their destination. Obviously, we cannot transport the commodities $1, 2$ and 3 from their origin to their destination by visiting all terminals only once, even if we use different vehicles.

We can highlight this impossibility by using a precedence graph constructed as follows. Each transportation service can be seen as a task. If the same commodity is carried on successive services, the first transportation service has to be done first and so on for the next transportation services. Let us construct the precedence graph of these different commodity tasks, we denote by T_{AB} the task "transportation service from A to B".

- The first commodity implies the following precedences : $T_{DA} \xrightarrow{1} T_{AB} \xrightarrow{1} T_{BC}$
- The second gives precedence : $T_{AB} \xrightarrow{2} T_{BC} \xrightarrow{2} T_{CD}$
- For the third we have : $T_{CD} \xrightarrow{3} T_{DA}$
- The other commodities do not imply any precedence, because they are transported directly.

Thus we need to schedule the task of precedence graph G represented in Figure 7.

$$\begin{tikzcd} T_{AB} \arrow[r, "1{,}2"] & T_{BC} \arrow[r, "2"] & T_{CD} \arrow[r, "3"] & T_{DA} \arrow[lll, bend left, "1"] \end{tikzcd}$$

Figure 7: Precedence graph on transportation tasks.

This graph has a cycle : it is impossible to schedule these tasks by respecting all time precedences. One solution could be to store some commodities in a location and for these to be delivered in the repeat of the transportation planning in the next time period. However, commodities would not arrive on time, so this solution is unacceptable in our study.

So we need to enhance the existence of cycles in the precedence graph of transportation tasks. To achieve this, we have to revise the choices we made when filling the network with commodities. From the invalid solution (with cycles), we generate new binary variables and constraints (Constraints 13 - 15) to prevent production of cycles. These new variables and constraints are easily added to multicommodity flow formulation. Once they are added, the modified multicommodity flow problem is solved again.

How to find and eliminate time incoherences To know which commodities are responsible for the appearance of a cycle and on which transportation arcs, we construct a directed graph G' as follows. The vertices of this graph are the transportation demands k, the arc (k, k') means that demand k is in conflict with demand k'. In the precedence graph G, we consider that demand k is in conflict with demand k' if and only if k' is loaded after k and k is unloaded before k' at the end of an operation belonging to G.

Proposition 4.1 *The graph G' has a cycle if and only if the precedence graph G has a cycle.*

We can identify the demands which involve the cycles in G' and, resulting from our proposal, these demands are also responsible for the cycle in the precedence graph G. We detect the cycles in G and G' with Floyd's algorithm. These demands will be called the incriminated demands. We will introduce new constraints and refill the network with commodities (multicommodity flow problem with some added constraints) so as to discard at least one precedence between transportation tasks, and to eliminate at least one cycle in the precedence graph. This constraint will impact the transportation arcs on which at least two incriminated demands are transported. It is clear that the arcs at the end of the transportation of incriminated demands appear in the cycle and carry more than one demand. Thus, we introduce constraints on these arcs.

Let $\{k_1, k_2, \ldots, k_{m+1} = k_1\}$ be a cycle in G'. The last transportation arcs in the graph G shared by the demands k_i and k_{i+1} (involved in this cycle) are denoted as a^{k_i}. As we need to discard the cycle we have to reroute at least one commodity, in other words, we need to choose which commodities will keep the same route and which will not. Thus we introduce variable u^k_{ij} which determines whether or not commodity k will be carried on transportation arc (i, j). We need these variables only for the demands involved in the cycle. The added constraints are as follows:

$$x^{k_i}_{a^{k_i}} \leq w^{k_i} \cdot u^{k_i}_{a^{k_i}} \qquad \forall i \in [1..m] \tag{13}$$

$$x^{k_{i+1}}_{a^{k_i}} \leq w^{k_{i+1}} \cdot u^{k_{i+1}}_{a^{k_i}} \qquad \forall i \in [1..m] \tag{14}$$

$$\sum_{i \in [1..m]} (u^{k_i}_{a^{k_i}} + u^{k_{i+1}}_{a^{k_i}}) \leq 2m - 1 \tag{15}$$

$$u^{k_i}_{a^{k_i}} \in \{0, 1\} \qquad \forall i \in [1..m] \tag{16}$$

$$u^{k_{i+1}}_{a^{k_i}} \in \{0, 1\} \qquad \forall i \in [1..m] \tag{17}$$

Note that l has disappeared in the index of x variables because fleets are assigned to arcs. Constraints (13) and (14) guarantee that commodity k_i is carried on transportation arc a^{k_i} only if this arc is kept for it (that is if $u^{k_i}_{a^{k_i}} = 1$). The constraint (15) forces that at least one conflict between 2 demands belonging to cycle in G' will be canceled, i.e. one demand will not share the transportation arcs with the other conflicting demands. One commodity will change its transportation path. Thus after constraint generation the incriminated demands will not all follow the same paths as before, and at the very least the cycle $\{k_1, k_2, \ldots, k_{m+1}\}$ in G' will be eliminated. So the incriminated demand will no longer produce the same precedences in G.

Example Let us see how this looks on our example. The three commodities 1, 2 and 3 are responsible for the cycle in the precedence graph above. These commodities follow the following paths respectively: *DA–AB–BC*, *AB–BC–CD* and *CD–DA*. In order to break the cycle in the precedence graph, at least one of the these commodities has to follow another path or even not be carried. The graph of incriminated demands G' (Figure 8) is represented below, and the critical transportation arcs are specified over the arcs of G'.

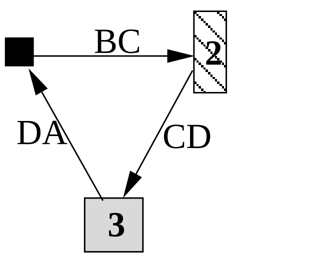

Figure 8: Graph of incriminated demands.

$$\begin{aligned}
&x^1_{BC} \leq w^1 \cdot u^1_{BC}\\
&x^1_{DA} \leq w^1 \cdot u^1_{DA}\\
&x^2_{BC} \leq w^2 \cdot u^2_{BC}\\
&x^2_{CD} \leq w^2 \cdot u^2_{CD}\\
&x^3_{CD} \leq w^3 \cdot u^3_{CD}\\
&x^3_{DA} \leq w^3 \cdot u^3_{DA}\\
&u^1_{BC} + u^2_{BC} + u^2_{CD} + u^3_{CD} + u^3_{DA} + u^1_{DA} \leq 5
\end{aligned}$$

Figure 9: Constraints to involve cycles on the incriminated demands

The constraints we added are detailed in Figure 9. These new constraints enforce one of the three incriminated demands to not be loaded, or to be rerouted if other transportation services not represented here are available. Thanks to these new constraints, the precedence graph no longer has a cycle dealing with these demands on the specified arcs.

How to more efficiently eliminate time incoherences The elimination of time incoherences may take a long time and we cannot ensure that it is done in polynomial time. Indeed, we have binary variables, so there is no time guarantee. The number of introduced variables is limited by the number of commodities ($|K|$) and the size of the network. However, we have no guarantee that the number of constraints (15) we introduce is polynomial : in some iterations no new variables may be introduced but a new constraint (15) may be added.

So as to speed up this step, we tried to discard more than one cycle before refilling arcs with commodities. We used an aggregate formulation which allows the reduction of the number of introduced binary variables and, consequentially, the number of constraints (15). Thus, discarding a cycle in this aggregate version allows us to eliminate the development of a lot of similar cycles at the same time. Moreover, so as to eliminate more time incoherences at each step, we try to find several cycles in the graph G' and in the precedence graph at each step. Even with these improvements, we have no time guarantee. We will see in Section 5 how often elimination of time incoherences prevents us from getting a solution in a reasonable time.

4.3 Construction of the vehicle plannings

It is only in this step that we care about time constraints. If transhipment is considered, these constraints are becoming increasingly difficult because of precedences between transportation arcs : if the same commodity is transported on successive arcs, then these arcs have to be scheduled in the same order.

4.3.1 Constraint Programming

The work has to be distributed evenly between the different vehicles over one planning horizon : any used vehicle l should be used at least $Quota^l_{min}$ but less than $Quota^l_{max}$. Thus in a linear model variables would have to be indexed by the vehicles (resources), and construction of the vehicle planning would be neither quick nor easy. This problem can be seen as a graph decomposition problem. This problem is detailed in [Teypaz and Rapine, 2008]. Similar problems have been proved NP-Complete [Dor and Tarsi, 1997].

If we consider an Eulerian cycle, a simple algorithm can solve the problem if the chosen Eulerian cycle allows it : when the cycle can be decomposed into paths with lengths between $Quota^l_{min}$ and $Quota^l_{max}$. This technical argument is the first one which made us tend towards Constraint Programming. Indeed, if the chosen Eulerian cycle does not bring a solution, we may branch to another alternative for the construction of this cycle. The second argument for using Constraint Programming is that it makes it easier to consider some secondary constraints for planning problem such as : transit times, regular departure times (vehicle departure time from a given location should always be the same), unavailability times (a vehicle cannot arrive or start at certain times of the day).

Constraint Programming methods are not only simple backtracking algorithms. In [Kumar, 1992], different techniques are proposed to improve backtracking. These techniques are : constraints propagation in a node of the search tree, intelligent backtracking in order to avoid repetition of identical failure schemes and the order in which variables are instantiated. To solve our problem using Constraint Programming we have to determine successions of transport sections. We use GNU-Prolog compiler [Diaz, 2007] which is a constraint solver on finite domains.

4.3.2 Construct the vehicle plannings using Constraint Programming

Construction of vehicle plannings can be seen as the following scheduling problem : each transportation service is a task whose duration is the transportation time, vehicles are resources which execute these tasks, some tasks have to be scheduled before others (precedence constraints on tasks). Contrary to most scheduling problems not all tasks can be scheduled after a given task. Two consecutive tasks on a resource must verify that the arrival terminal of the transportation service corresponding to the first task is also the departure terminal of the service corresponding to the second task (geographical constraints).

In practice, at each step of the search, we schedule one task, which instantiates one variable. The domain of time variables of each task (indicating the start of the tasks) are updated in function of the graph of precedence. At each node of the search tree, a task is scheduled either on the current resource or on a new one. latter case instantiates a new variable which uses new resources. We introduced different rules corresponding to these instantiations.

The priority is to schedule the task as a part of the chosen Eulerian cycle (Euler's algorithm) on the current resource. This is possible until $Quota_{max}$ is reached, in which case we use a new resource. At the end a failure may happen, i.e. $Quota_{min}$ is not respected. In that case we have to backtrack, and either to change the used rule or to follow another Eulerian cycle. If transhipment is allowed, we introduce new rules which allows us to consider precedences between transportation tasks in priority.

The order in which we apply the different rules is essential to arrive at a solution. Nevertheless, it may happen that no feasible solution respecting time quotas and delivering all commodities in the same planning horizon, exists or can be computed in reasonable time.

5 Tests and results

We present results obtained using an homogeneous fleet of identical vehicles. Some remarks on the heterogeneous fleet will also be given.

5.1 Instances

Our methods were tested on instances provided by our industrial partner. The solutions obtained give some improvements: there is little increase in profit, but the fleet size is reduced significantly. To test further while preserving data confidentiality, an instance generator based on data studies from a real freight transportation problem were used. The instances contain 40 terminals and we know distances $d(i,j)$ between the different terminals. Our time period is of 480 time units, during such a period a used vehicle should be used at least for $Quota_{min}$ time units and at most $Quota_{max}$ time units (for the most profitable vehicle, considered in most of the tests, we choose $Quota_{min} = 130$ and $Quota_{max} = 325$). We have different vehicle types l ($l = 1$ or 3), and know their capacities C^l, the capacity of the most profitable vehicle is of 140 transport units. For any type l of vehicle, the fixed cost is given by matrix (F^l_{ij}), and the variable cost is given by (U^l_{ij}) (which is the coefficient of the linear function $F^l_v(W,i,j)$). We also know transportation times between all pairs of terminals, detailed in matrix (t^l_{ij}).

We choose different percentages of filling for the demands matrix and demands are either concentrated at certain points, which we may call hubs, or not. The four different ways of filling the demands matrix are:

- Demands are concentrated on one point (hub), that means the row and the column corresponding to the hub are filled at 80% with non zero demands, and that the remainder is filled at 5% (this corresponds to 8.75% non zero demands).
- The matrix is filled randomly with 8.75% of non zero demands.
- Demands are concentrated on 3 hubs, (this corresponds to 16% non zero demands).
- The matrix is filled randomly with 16% of non zero demands.

In all cases, the filling is done such that all terminals are involved in at least one non-zero demand, as origin or destination. The analysis of real data allows to model how transportation prices depend on distance. We used it to compute the fixed transportation prices.

The size of any demand is either small compared to vehicle size, or equivalent to it or big compared to vehicle size (Table 1).

W^k	1-30	30-90	90-150	150-250	250-800
Distribution					
Small Demands	40%	40%	10%	5%	5%
Equivalent Demands	5%	20%	50%	20%	5%
Big Demands	5%	5%	10%	40%	40%

Table 1: Distribution of the size of any demand.

Thus we have 12 different profiles of instances. We tested all of these profiles with both algorithms (with and without transhipment) and for both methods of construction of the network (heuristic and linear relaxation). For each profile, we generated 10 instances and shown is the mean of the results obtained.

To evaluate the financial performance, we compare our results to a lower bound obtained with direct trip policy (after first step of our heuristic for construction of the network) and the upper bound obtained using Service Network Design formulation (Section 2.3). So as to compare the different methods more precisely we detail the use of resources (time and vehicles). Additionally, the time necessary to obtain a solution and the number of vehicles used are both important to an evaluation of solutions.

5.2 Resolution time and failures

Our industrial partner required our method to be able to solve their instances in less than 1 hour. Thus, to test all our methods, we limited the time to 20 minutes for solving an instance. In Table 2, for both methods of constructing the planning (first line), the first two columns summarize the mean resolution time for the method without transhipment (NoT) and with transhipment (T). The next two columns specify the number of failures of methods with transhipment : the number of times that the resolution was stopped during the demand-related filling step (step2) is detailed in column (Fail2), the number of times that the resolution was stopped during construction of the planning (step3) is summed up in column (Fail3). Fail2 is due to too many time incoherences, Fail3 is caused by too many difficult precedence constraints on transportation services. These two failures are of a different nature, in the second case we already know the global profit, construction of planning is only a decision problem.

Instances	Direct Trip Policy				Linear Relaxation			
	NoT Time	T Time[1]	Fail2	Fail3	NoT Time	T Time[1]	Fail2	Fail3
Small Demands < vehicle size								
1 hub	0,32	1,25	0	2	61,83	62,67	0	2
3 hubs	0,44	40,26	0	3	71,35	151,20	0	3
8,75%	0,33	22,33	0	1	58,34	127,36	1	2
16%	0,43	65,18	4	1	87,71	130,25[2]	9	0
Demands ≈ vehicle size								
1 hub	0,46	0,88	0	1	64,10	62,73	0	3
3 hubs	0,60	29,44	0	1	73,59	83,97	0	1
8,75%	0,44	10,94	0	0	62,79	132,59	3	0
16%	0,62	209,30	0	1	78,58	218,83	5	1
Big Demands > vehicle size								
1 hub	0,78	1,95	0	2	62,09	74,33	0	0
3 hubs	1,47	2,83	0	2	84,27	177,80	0	0
8,75%	0,73	2,15	0	2	66,20	66,79	0	1
16%	1,48	7,95	0	1	81,06	116,12	0	0
total			4	17			18	13

Table 2: Computation time (in seconds) and resolution failures.

Methods without transhipment are very robust, they provide a solution for all instances and in approximately 1 second if direct trip policy is used for network design, and 1 minute with linear relaxation for network design. With transhipment, we do not always have a solution. When the network is constructed using linear relaxation, computation time is longer and failures are even more frequent. Furthermore, we cannot measure the gains due to transhipment, because computation fails too often during step 2.

5.3 Solution results

5.3.1 Optimization

First, we present the financial results of the different methods compared to the lower and upper bounds (see Table 3). The lower bound gap decreases as the size of demands increases. This is mainly due to the scale of objective (millions or billions) which increases when the size of demands increase, and to the fact that the difference between upper and lower bound is nearly constant. The last column gives the relative gain of the best solution obtained by ILOG Cplex in 20 minutes.

So, to compare the different values, we calculate the relative gain (RG), which is the difference between the result and the lower bound, divided by the difference between the upper and the lower bound. We expressed this relative gain as a percentage. A negative relative gain means that the profit is worse than the lower bound, a relative gain of 100% means that the profit is equal to the upper bound. To synthesize results from the 12 profiles, average and standard deviation σ are given at the end of the table.

We also specify the percentage of satisfied demands in the columns labeled "Sat".

Instances	Lower Bound Gap	Direct Trip Policy				Linear Relaxation				Cplex
		NoT		T		NoT		T		
		RG	Sat	RG	Sat	RG	Sat	RG	Sat	RG
Small Demands/ vehicle size										
1 hub	34,85	12,63	68,19	31,96	69,11	-2,41	68,70	31,46	71,92	81,31
3 hubs	34,78	18,59	72,86	49,01	76,06	6,59	73,75	57,66	79,32	90,21
8,75%	36,59	5,83	64,74	25,56	64,53	-11,73	64,40	19,23	65,76	81,67
16%	34,53	10,36	69,95	40,95[1]	72,34	-7,78	73,16	50,81[2]	83,03	90,36
Demands $\approx$ vehicle size										
1 hub	15,27	16,68	86,99	42,33	88,57	-7,49	84,94	43,25	87,70	79,93
3 hubs	15,82	16,81	86,41	51,15	89,27	3,32	84,72	68,26	88,29	89,64
8,75%	15,87	7,29	80,74	37,92	81,68	-17,91	81,44	33,86[3]	82,66	79,48
16%	15,97	10,46	83,06	47,82	84,63	-12,87	82,43	63,36[4]	86,60	89,84
Big Demands/ vehicle size										
1 hub	8,50	16,38	87,26	41,63	88,23	-4,26	86,68	45,87	87,55	83,10
3 hubs	8,15	21,88	87,78	53,15	88,74	7,83	87,51	70,21	88,74	90,30
8,75%	8,35	8,55	86,87	36,84	86,95	-14,55	87,02	39,38	87,75	82,92
16%	8,43	13,59	88,21	48,91	86,95	-7,82	88,46	61,98	89,66	90,65
Average		13,25		42,31		-5,76		48,61		85,78
σ		5,78		9,41		11,95		17,26		5,00

Table 3: Relative gain and percentage of satisfied demands.

We can see the effect of hubs: hubs allow an increase in profit compared to the lower bound. Transportation links are essentially concentrated on the hubs, thus efficient networks are easier to construct. This is confirmed by a higher rate of satisfaction of demands.

[1] If resolution fails, its computation time is not used to compute the average.
[2] Only one result.
[1] Average of 6 results,
[2] Only one result,
[3] Average of 7 results,
[4] Average of 5 results.

The method without transhipment and based on a network construction using linear relaxation is often worse than the lower bound. Indeed, linear relaxation constructs networks privileging transhipment ; this prevents construction of independent plannings and less profitable commodities are carried.

Transhipment allows us to increase profit. With transhipment, the profits are nearer to the upper bounds. Network construction using linear relaxation is very efficient for instances containing hubs. for instances containing hubs. However, when demands are randomly generated (less structured), of small or medium size, the use of direct trip policy is more attractive. The profits are higher, and furthermore, this method provides a solution for nearly all instances. Also, the method is less influenced by the profile of the instance (the standard deviation σ is lower).

5.3.2 Vehicle utilization

For different methods we calculated the average utilization time of the vehicles (Table 4). These times are similar for the different profiles of instances.

Instances	Direct Trip Policy		Linear Relaxation	
	NoT	T	NoT	T
Average utilization time	291,3	188	292,7	187,5

Table 4: Vehicle utilization.

Without transhipment, vehicle utilization times are close to $Quota_{max} = 325$, thus vehicles are almost optimally used. On the other hand, the methods with transhipment are not using the vehicles efficiently, utilization times are close to $Quota_{min} = 130$. In spite of respect of time quotas and a better profit, methods with transhipment are unsatisfactory : they use 1.5 times as many vehicles than if no transhipment is done. This is due to the fact that there is no fixed cost on the vehicles.

The method without transhipment using direct trip policy for network construction is the more reliable. Computation is very fast (some seconds), resources are well used and the profit is higher than the lower bound. Methods with transhipment get better profits, but vehicles are under-used. Furthermore, 20 minutes of resolution time is not always enough to find a solution.

5.4 Heterogeneous fleet

We tested the same instances with an heterogeneous fleet containing the same type of vehicle as homogeneous fleet and two other types with smaller capacities (100 and 60), which are competitive compared to the bigger vehicle especially for small demands. The network constructed by using linear relaxation is nearly the same as for the homogeneous fleet, but as the linear model is bigger, it takes a longer time to obtain the network.

The lower bound is improved, and also the results for small demands but the benefit is quite small. Indeed, for medium and high demands, using an heterogeneous fleet gives worse results ! The networks constructed for small vehicles (positive cycles) often have no commodities to carry when filling them (the cycle becomes negative). These commodities are taken by larger vehicles.

6 Conclusion and perspectives

We studied a freight transportation problem which consists of maximizing the carrier profit. It is not necessary to totally satisfy all demands but profitable demands should be satisfied globally or partially. The carrier must use his vehicles better, some repositioning of vehicles with empty load may be needed in order to complete the vehicle rotation, the vehicles must not be under- or over-used. To improve the profit, the carrier can use transhipment operations. The vehicle fleet can be adjusted with different types of vehicle or different modes of transport. For our solving approach, we need to design the routes for each satisfied demand and the planning of vehicles which carry all satisfied demands.

Two solving methods were studied for problems with and without transhipment. We decomposed the freight transportation problem into three main steps. This enables us to take all parameters of the problem and also specific time constraints into account. All of these steps have been studied in detail. They also involved graph algorithms (flow problems), as Linear and Mixed Integer Programming and Constraint Programming techniques. The first step of our resolution is critical for profit. In order to improve profit, other methods for the network construction will be investigated. The filling step is optimal for its inputs according to the two methods, however the need to eliminate time incoherences means that obtaining a solution in a reasonable time cannot be guaranteed. The vehicle planning construction is efficient for the transhipment method in spite of the problem being probably NP-Complete. The precedences constraints complicate planning construction and Constraint Programming often fails.

Our solving method which does not consider transhipment guarantees a rapid solution (some seconds) and for sure a solution rapidly with an improvement in profit compared to lower bound. The profit is significantly increased by allowing transhipment, and even though utilization of time quotas are respected, vehicles are not well used. Computation times stay fairly low (about one minute) when precedence constraints for the planning problem are relatively easy to satisfy. These results are quite encouraging, and are often more efficient than the method based on linear relaxation, which fails more often. Nevertheless, in our method with transhipment, computing the transportation planning is not always done in an acceptable time.

In this problem we did not take the fleet size into account, which may not be easily integrated in our resolution scheme. This new constraint should already be considered within the network construction. Furthermore, this constraint also has to be taken into account during the planning step, and our resolution may fail, especially when transhipment is allowed (explosion of computation time due to precedences on transportation arcs and to constraints on fleet size). A formulation of our problem based on routes (in this case, a route is a complete planning of a vehicle) combined with Service Network Design formulation may be more suitable for this problem. These aspects are still under investigation.

Aknowledgments

This research has been supported by the following grants and contracts: BQR INPG "Optimisation du transport de fret par l'utilisation de plateformes logistiques"; Cluster de Recherche GOSPI, Gestion et Organisation des Systèmes de Production et de l'Innovation, Région Rhônes-Alpes.

References

[Ahuja et al., 1989] Ahuja, R., Magnanti, T., and Orlin, J. (1989). *Optimization*, chapter Network flows, pages 211–369. Elsevier North-Holland, Inc.

[Andersen et al., 2007a] Andersen, J., Christiansen, M., Grønhaug, R., and Crainic, T. (2007a). Branch-and-price for service network design with asset management constraints. Technical Report 55, Centre Interuniversitaire de Recherche sur les Réseaux d'Entreprise, la Logistique et le Transport, Canada.

[Andersen et al., 2007b] Andersen, J., Crainic, T., and Christiansen, M. (2007b). Service network design with asset management: Formulations and comparative analyses. Technical Report 40, Centre Interuniversitaire de Recherche sur les Réseaux d'Entreprise, la Logistique et le Transport, Canada.

[Andersen et al., 2007c] Andersen, J., Crainic, T., and Christiansen, M. (2007c). Service network design with management and coordination of multiple fleets. *European Journal of Operational Research*.

[Barnhart et al., 2000] Barnhart, C., Jin, H., and Vance, P. (2000). Railroad blocking : a network design application. *Operations Research*, 48(4):603–614.

[Barnhart et al., 2002] Barnhart, C., Krishnan, N., Kim, D., and Ware, K. (2002). Network design for express shipment delivery. *Computational Optimization and Applications*, 21(3):239–262.

[Barnhart and Schneur, 1996] Barnhart, C. and Schneur, R. (1996). Network design for express freight service. *Operations Research*, 12(6):852–863.

[Chou et al., 2003] Chou, M., Song, M., and Teo, C. (2003). Inventory-routing problem in sea freight: Direct versus transshipment model. Technical report, The Logistics Institute - Asia Pacific.

[Crainic, 2000] Crainic, T. (2000). Service network design in freight transportation. *European Journal of Operational Research*, 122(2):272–288.

[Crainic, 2002] Crainic, T. (2002). *Handbook of Transportation Science*, chapter Long-Haul Freight Transportation, pages 451–516. Kluwer's International Series.

[Diaz, 2007] Diaz, D. (2007). GNU Prolog: A Native Prolog Compiler with Constraint Solving over Finite Domains.

[Dor and Tarsi, 1997] Dor, D. and Tarsi, M. (1997). Graphe decomposition is np-complete : A complete proof of holyer's conjecture. *SIAM Journal on Computing*, 26(4):1166–1187.

[Fleischer, 2000] Fleischer, L. (2000). Approximating fractional multicommodity flows independent of the number of commodities. *SIAM Journal on Discrete Mathematics*, 13(4):505–520.

[Garg and Konemann, 1998] Garg, N. and Konemann, J. (1998). Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *39th Annual IEEE Symposium on Foundations of Computer Science*, pages 300–309.

[Hahgani and Oh, 1996] Hahgani, A. and Oh, S. (1996). Formulation and solution of a multi-commodity, multi-modal network flow model for disaster relief operations. *Transportation Research Part A*, 30(3):231–250.

[Jaillet et al., 1996] Jaillet, P., Song, G., and Yu, G. (1996). Airline network design and hub location problems. *Location Science*, 4(3):195–212.

[Kumar, 1992] Kumar, V. (1992). Algorithms for constraint satisfaction problems: A survey. *AI Magazine*, 13(1):32–44.

[Lai and Lo, 2004] Lai, M. and Lo, H. (2004). Ferry service network design: optimal fleet size, routing, and scheduling. *Transportation Research Part A*, 38:305–328.

[Laporte, 1992] Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(3):345–358.

[Pedersen et al., 2007] Pedersen, M., Crainic, T., and Madsen, O. (2007). Models and tabu search meta-heuristics for service network design with asset management requirements. 49, Centre Interuniversitaire de Recherche sur les Réseaux d'Entreprise, la Logistique et le Transport.

[Smilowitz et al., 2003] Smilowitz, K., Atamtürk, A., and Daganzo, C. (2003). Deferred item and vehicle routing within integrated networks. *Transportation Research Part E*, 39:305–323.

[Teypaz and Rapine, 2008] Teypaz, N. and Rapine, C. (2008). Graph decomposition into paths under length constraints. Technical Report 164, Cahiers de Leibniz - G-SCOP.

[Wieberneit, 2008] Wieberneit, N. (2008). Service network design for freight transportation: a review. *OR Spectrum*, 30:77–112.

[Yan et al., 2005] Yan, S., Chen, S., and Chen, C. (2005). Air cargo fleet routing and timetable setting with multiple on-time demands. *Transportation Research Part E*, 42(5):409–430.

Les cahiers Leibniz ont pour vocation la diffusion des rapports de recherche, des séminaires ou des projets de publication sur des problèmes liés au mathématiques discrètes.

Pour soummettre un articles dans les cahiers,
`http://www.g-scop.inpg.fr/CahiersLeibniz/`

Responsables de la publication : Nadia Brauner et András Sebő
ISSN : 1298-020X -