Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

# Service Network Design with Asset Management: Formulations and Comparative Analyzes


**Jardar Andersen**
**Teodor Gabriel Crainic**
**Marielle Christiansen**



**October 2007**

**CIRRELT-2007-40**



**Bureaux de Montréal :**
Université de Montréal
C.P. 6128, succ. Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

**Bureaux de Québec :**
Université Laval
Pavillon Palasis-Prince, local 2642
Québec (Québec)
Canada G1K 7P4
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

# Service Network Design with Asset Management: Formulations and Comparative Analyzes

**Jardar Andersen[1,2,*] ,Teodor Gabriel Crainic[3], Marielle Christiansen[1]**

[1] Department of Industrial Economics and Technology Management, Norwegian University of Science and Technology, 7491 Trondheim, Norway

[2] Institute of Transport Economics, Gaustadalléen 21, 0349 Oslo, Norway

[3] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT), Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Canada H3C 3J7 and NSERC Industrial Research Chair in Logistics Management, Université du Québec à Montréal, C.P. 8888, succursale Centre-ville, Montréal, Canada H3C 3P8

**Abstract.** In this paper, we address the Service Network Design with Asset Management problem, SNDAM. This problem arises when vehicles need to be considered while designing service networks, and integrates a new layer of vehicle management decisions to the service network design problem that traditionally has considered selection of services (design arcs) and routing of flow. We introduce extended asset management constraints to the problem and propose alternative formulations of the SNDAM problem based on new cycle design variables. In the computational study, we solve test cases representing real planning problems in order to analyze the strengths and weaknesses of the various model formulations and the impact of asset management considerations on the transportation plan. Experimental results indicate that formulations based on cycle variables outperform traditional arc-based formulations, and that considering asset management issues may significantly impact the outcome of the service planning models.

**Keywords**. Service network design, freight transportation, capacitated multicommodity network design, asset management.

**Acknowledgements.** This work has received financial support from The Norwegian Research Council through the Polcorridor Logchain project. Partial funding has also been supplied by the Natural Sciences and Engineering Research Council of Canada (NSERC) through its Discovery and Industrial Research Chair grant programs.



---

* Corresponding author: jardar.andersen@iot.ntnu.no

Dépôt légal – Bibliothèque nationale du Québec,
Bibliothèque nationale du Canada, 2007

# 1 Introduction

Transportation is among the most vital industries in a modern economy. Increased competition and reduced profit margins in the transport industry have increased the need for efficient operations for carriers. Simultaneously, there has been a change in attitude among customers with increased focus on service quality. Service providers in the transport industry are thus facing a more challenging situation than just a few decades ago, and survivability is becoming more dependent of intelligent planning, design, and execution of operations. For this purpose, decision support systems and optimization models have been increasingly used to assist in decision-making and resource allocation.

There has traditionally been a distinction between three planning levels in transportation planning (Crainic and Laporte, 1997, Macharis and Bontekoning, 2004). Strategic planning deals with long-term issues (several years) like major investments and general development policies. Examples of strategic planning issues are location of terminals, network configurations, and design and layout of terminals. Tactical planning covers medium-term issues (typically weeks, months) such as efficient allocation and utilization of resources. One example is service network design covering issues like determination of service schedules and routing of flow. Operational planning covers short-term issues (real-time, days, weeks) in a dynamic environment where the time factor is particularly important. Examples are implementation and adjustment of schedules for services and dynamic allocation of scarce resources.

*Service network design* models address issues like route selection and terminal policies, whether to use direct or consolidating services, and how often to operate (Crainic, 2000). Service network design issues are usually considered to be at a tactical planning level, but aspects of strategic or operational planning may also be considered. The resulting formulations are often fixed charge capacitated multicommodity network design (CMND) models. Services are represented as arcs or paths in the network that may be operated or not. The fixed charges represent costs that must be paid for each network arc that is opened, while the network is capacitated because several flows need to share capacities of the arcs that are opened. In order to meet requirements for more intelligent planning, service network design models are increasingly being made more sophisticated. Simultaneously, advances in computational capacity and methodological development have increased the range of problems that can be solved.

Traditional service network design studies have included two layers: The *design* layer includes decisions on what services to operate, while the *flow* layer focuses on routing the demand in the system. From the application side, a need for inclusion of a third layer to service network design model has been expressed, namely a layer representing *assets* that are needed to operate services. The standard approach has however been to manage such assets in an *a posteriori* analysis. Vehicles that are needed for operation of transport services are an example of such assets. When assets are considered in the service network design process, we refer to the resulting models as *Service Network Design with Asset Management* (*SNDAM*). In this paper, we study the impact of asset management on service network design.

One asset management issue that *has* been studied in the literature is ensuring that there is an equal number of assets entering and leaving each terminal (node) in the network. Constraints ensuring this are referred to as design balance constraints (Pedersen, 2006; Pedersen et al., 2007). Design balance constraints have been modeled for various modes of transportation, see e.g. Smilowitz et al. (2003), Lai and Lo (2004), Barnhart and Schneur (1996), Kim et al. (1999) and

Andersen et al. (2007). However, these constraints and their impact on the service network design models and plans have not been extensively studied in the literature, and most other asset management issues have not been addressed at all. This paper aims to contribute to fill this gap.

The goal of this paper is to discuss how service network design models can incorporate various asset management issues, and to analyze how the explicit consideration of asset management issues impacts the solutions that are produced by service network design models. We present four alternate formulations of an extended model by introducing cycles as design variables in addition to the more common path variables for the flow. We compare experimentally the four model formulations on problem instances that can be handled through a priori enumeration of columns (cycles and paths). The same set of problem instances is used as a test bed for evaluating the impact of constraints representing various asset management issues.

The contributions of the paper are thus twofold. First, it offers a comprehensive modeling framework for SNDAM, together with cycle and path generation methods exploring the special structure of the formulations. Second, the paper contributes with an extended analysis of how asset management issues affect service network design.

The outline of the paper is as follows. In Section 2, we recall service network design and associated earlier work on asset management. Section 3 presents an extended model for SNDAM. We develop three alternate formulations of the extended SNDAM model in Section 4, and present algorithms for model solving in Section 5. The computational study, comparing the different formulations of the extended model and analyzing the impact of asset management issues on service design, is presented in Section 6. Finally, concluding remarks are given in Section 7.

# 2 Service network design formulations

In this section, we recall formulations of service network design from the literature. We introduce the fixed charge capacitated multicommodity network design (CMND) problem in Section 2.1, CMND being the traditional formulation for service network design problems. In Section 2.2, we recall the basic formulation of the Service Network Design with Asset Management (SNDAM) problem, incorporating the first asset management issue considered in the service network design literature.

## 2.1 Fixed charge capacitated multicommodity network design

The directed graph $\mathrm{G} = (\mathrm{N}, \mathrm{A})$ represents the network, where $\mathrm{N}$ is the set of nodes and $\mathrm{A}$ the set of arcs. Without loss of generality, we assume that all arcs $(i, j) \in \mathrm{A}$ are design arcs. In this paper, we refer to commodities as products $\mathrm{P} = \{p\}$ that need to be transported through the network. Each product has a demanded volume $w^p$ that has to be transported from the unique origin node of the product $o^p$ to its unique destination node $d^p$.

Design variables $y_{ij}$ are binary variables indicating whether an arc is used or not, while the flow variables $x_{ij}^p$ are nonnegative real numbers. Each design arc $y_{ij}$ has an associated capacity $u_{ij}$, and a fixed cost $f_{ij}$ associated with the use of the arc. For each unit of product $p$, there is a flow cost $c_{ij}^p$ for traversing arc $(i, j) \in \mathrm{A}$. Sets $\mathrm{N}^+(i) = \{ j \in \mathrm{N} : (i, j) \in \mathrm{A} \}$ and $\mathrm{N}^-(i) = \{ j \in \mathrm{N} : (j, i) \in \mathrm{A} \}$ of outward and inward neighbors are defined for each node. Let $b_{ij}^p = \min\{ w^p, u_{ij} \}$. The problem consists of minimizing the sum of fixed costs and flow costs while satisfying all demand, and the arc-based formulation of the CMND problem is formulated in (1) – (5).

$$Min\ z = \sum_{(i,j)\in A} f_{ij} y_{ij} + \sum_{(i,j)\in A} \sum_{p\in P} c_{ij}^{p} x_{ij}^{p} \tag{1}$$

$$\sum_{j\in N^{+}(i)} x_{ij}^{p} - \sum_{j\in N^{-}(i)} x_{ji}^{p} = \begin{cases} w^{p}, & i = o^{p} \\ -w^{p}, & i = d^{p} \\ 0, & otherwise\ , \end{cases} \qquad \forall i \in N, p \in P, \tag{2}$$

$$\sum_{p\in P} x_{ij}^{p} - u_{ij} y_{ij} \le 0\ , \qquad \forall (i,j) \in A, \tag{3}$$

$$x_{ij}^{p} - b_{ij}^{p} y_{ij} \le 0\ , \qquad \forall (i,j) \in A, p \in P\ , \tag{4}$$

$$x_{ij}^{p} \ge 0, \qquad \forall (i,j) \in A, p \in P\ , \tag{5}$$

$$y_{ij} \in \{0,1\}, \qquad \forall (i,j) \in A. \tag{6}$$

The objective function (1) minimizes the sum of fixed costs for opening design arcs and flow costs for all products that are transported through the network. Constraints (2) are node balance constraints for the flow. Constraints (3) and (4) are weak and strong forcing constraints, respectively, defining arc capacities and forcing flows to zero when arcs are not opened. Constraints (4) are redundant in the MIP-formulation, but have appeared to improve the lower bounds obtained through relaxations (Crainic et al., 2001). The formulation including (4) is the strong formulation, while removal of (4) results in the weak formulation. Constraints (5) and (6) are variable-type constraints, restricting $x_{ij}^{p}$ to nonnegative real numbers and $y_{ij}$ variables to binary values.

An equivalent network design formulation is based on path variables for the flow (e.g., Ahuja et al., 1993; Crainic et al., 2000). We define the set of paths $\mathcal{L}^{p}$ that product $p$ may use from its origin node to its destination node. For these paths, we define parameters $a_{ij}^{pl}$ =1 if arc $(i, j) \in A$ belongs to path $l \in \mathcal{L}^{p}$ for product $p$, 0 otherwise. The flow of product $p$ on path $l$ is $h^{pl}$, while the flow cost for transporting product $p$ on path $l$ is $k^{pl}$, $k^{pl} = \sum_{(i,j)\in A} c_{ij}^{p} a_{ij}^{pl}$ . The path-based formulation can then be presented as in (7)-(12).

$$Min\ z = \sum_{(i,j)\in \mathrm{A}} f_{ij} y_{ij} + \sum_{p\in \mathrm{P}} \sum_{l\in \mathcal{L}^p} k^{pl} h^{pl} \tag{7}$$

$$\sum_{l\in \mathcal{L}^p} h^{pl} = w^p, \qquad \forall p \in \mathrm{P}, \tag{8}$$

$$\sum_{p\in \mathrm{P}} \sum_{l\in \mathcal{L}^p} a_{ij}^{pl} h^{pl} - u_{ij} y_{ij} \le 0, \qquad \forall (i,j) \in \mathrm{A}, \tag{9}$$

$$\sum_{l\in \mathcal{L}^p} a_{ij}^{pl} h^{pl} - b_{ij}^{p} y_{ij} \le 0, \qquad \forall (i,j) \in \mathrm{A}, p \in \mathrm{P}, \tag{10}$$

$$h^{pl} \ge 0, \qquad \forall p \in \mathrm{P}, l \in \mathcal{L}^p, \tag{11}$$

$$y_{ij} \in \{0,1\}, \qquad \forall (i,j) \in \mathrm{A}. \tag{12}$$

The objective function (7) minimizes the sum of fixed costs for opening design arcs and flow costs on paths. Constraints (8) now state that the sum of flow on all paths for a product has to equal the demand for this product. The other constraints are similar to those of (1)-(6), with the new variables and parameters substituted into the formulation: Strong and weak forcing constraints are found in (9) and (10), respectively, while variable-type constraints are presented in (11) and (12). We refer to model (1)-(6) as *aa*-CMND, because both design variables and flow variables are arcs. Model (7)-(12) is referred to as *ap*-CMND, because the flow variables are defined in terms of paths.

## 2.2 Basic SNDAM model

In this subsection we define the basic SNDAM model that corresponds to what was labeled *Design Balanced Multicommodity Network Design* in Pedersen (2006). The basic idea is that there should be an equal number of vehicles leaving and entering a node. We introduce design balance constraints (13) to ensure that all nodes have equal in- and out-degree of opened arcs.

$$\sum_{j\in \mathrm{N}^+(i)} y_{ij} - \sum_{j\in \mathrm{N}^-(i)} y_{ji} = 0, \qquad \forall i \in \mathrm{N}. \tag{13}$$

The balancing requirement expressed in (13) has been present in most design models within maritime and air transportation. For these modes of transportation, the focus is on routing expensive assets in an optimal way, which leads to the requirements for vehicle balance at terminals. Examples of such design studies with balancing requirements are Lai and Lo (2004) who model ferry service network design with balancing constraints for the ferries, and Barnhart and Schneur (1996), who model a balanced express shipment design problem for air transport.

The literature with design balance constraints is however sparse in traditional service network design. Smilowitz et al. (2003) present a model for multimodal package delivery with design balance constraints for ground vehicles, while Andersen et al. (2007) and Pedersen (2006; see also Pedersen and Crainic, 2007; Pedersen et al., 2007) present service network design models with design balance constraints for locomotives.

All the model formulations presented in Section 2 may be used on a static or a time-space network. In a time-space representation, the planning horizon is divided into a set of time periods, and the nodes in the static network are replicated in each time period. Service arcs represent movements in both time and space. In addition, *holding* arcs between consecutive time realizations of physical nodes represent the possibility that assets and flow wait at nodes. The time-space network increases the size of the network considerably, but allows for analysis of dynamic aspects within the planning horizon.

# 3 Extended SNDAM model

The basic SNDAM model presented in Section 2.2 represents an important step for improved representation of real-world decisions. There are however additional challenges from real-world problems that have to be addressed in service network design models. In this subsection we develop an extended SNDAM model with constraints representing several asset management issues integrated within the service network design. We introduce relevant definitions and notation in Section 3.1, while the different asset management issues are discussed in Section 3.2. The extended SNDAM model is presented in Section 3.3.

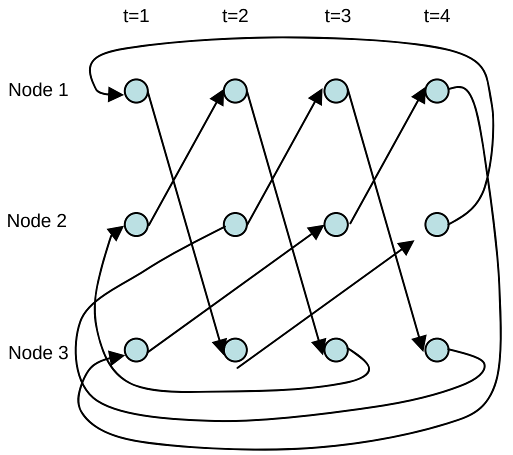
## 3.1 Definitions and notation

In this subsection, we present definitions and notation for the extended SNDAM model. We first discuss the time-space representation of network operations including holding and repositioning decisions, before presenting the notation in the second subsection.

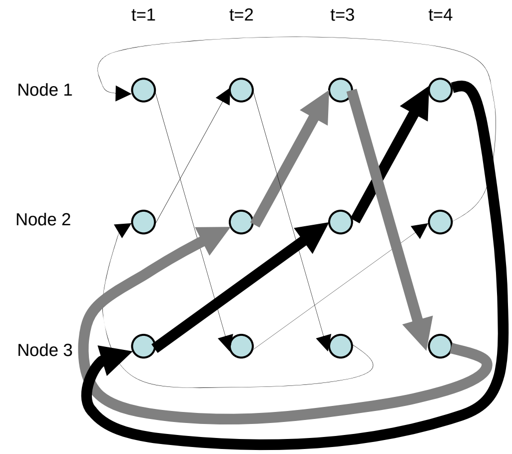
### 3.1.1 Time-space representation of network operations

The asset management aspects that we introduce require that we work in a time-space setting. In the rest of this paper we assume an initial static network, but use a time-space representation where the nodes in the static network are replicated in each time period. Figure 1 illustrates a time-space network with three physical nodes and four time periods.

The usual representation in service network design is to let services with intermediate stops between the origin and destination nodes be represented as paths in the network, while non-stop services are represented with single arcs. In this paper, we assume that all services can be represented as network arcs, as illustrated in Figure 1. This does not affect the generality of the results, but allows us to keep the presentation simpler than if we introduced service paths. The solid lines in Figure 1a) represent design arcs that may be chosen for operation. In Figure 1b), six design arcs are opened, and these satisfy design balance constraints (13), as all nodes have the same in- and out-degree.




1a)




1b)

*Figure 1. Time-space diagram for a network with three nodes and four time periods (1a) and example of feasible service plan (1b).*

In time-space networks, holding arcs represent assets or flow units kept at a physical node from one time period to the next and, thus, link consecutive representations in time of the same physical node. In the following, we assume that the models always have holding arcs for the flow, so that flow units may be kept at a physical node from one time period to the next. Holding arcs for assets should be included unless it is required that the assets should operate without any idle time. The latter is not usually an advantage, because it considerably restricts the management of the assets. Holding arcs are assumed to have infinite capacity both for assets and for flow. We do not complicate the model formulation with particular notation and variables for holding arcs. In the rest of this paper we label arcs in the time-space network that are not holding arcs as *movement* arcs.

We do not include repositioning arcs for the assets. Notice that the design balance constraints (13) ensure that an equal number of assets enter and leave all nodes. Thus, repositioning may be represented by a service without flow. The restriction imposed by this approach is that direct repositioning is only permitted between nodes that are connected with services. This does not impact the generality of the development, however, because including explicit repositioning arcs does not change the structure of the model (but makes the presentation significantly more complex).

### 3.1.2 Notation

We define a time-space representation by dividing the planning horizon into a set of time periods $\mathrm{T} = \{t\} \equiv \{1,..,T_{MAX}\}$. We refer to the graph $\mathrm{G}' = (\mathrm{N}', \mathrm{A}')$ for the static service network that is not time-expanded, and to the graph $\mathrm{G} = (\mathrm{N}, \mathrm{A})$ for the time-space network. A node $i' \in \mathrm{N}'$ in the static service network is referred to as a *physical* node, and has $T_{MAX}$ time realizations $i \in \mathrm{N}$ in the time-space network. Each node $i \in \mathrm{N}$ has an associated time period, $T_i \in \mathrm{T}$, and represents a physical node $N_i' \in \mathrm{N}'$. We allow services to leave at any time period. The service arcs $(i', j') \in \mathrm{A}'$ are thus available in $T_{MAX}$ realizations, one for each time period. The duration of arcs $(i, j) \in \mathrm{A}$ in the time-space network is the same for all realizations in time of the arcs in the static service network and is equal to $T_j - T_i$. The flow costs on arcs, $c_{ij}^p$, represent time costs. They are calculated as the product of the value of time for a product and the traversal time of the arc. The maximum number of vehicles available is $V_{MAX}$ .

Products have specific origin nodes in the time-space network, but no unique destination nodes, because no explicit delivery due-dates are considered. We introduce sink-destination nodes and zero-cost arcs from all time realizations of a physical node to the sink-destination node for the product considered. Notice that, because product arc costs generally are represented as time costs, this approach will result in bringing the products to their destinations as early as possible, even though there are no hard constraints representing due times. Delivery due dates for products may be incorporated, however, by increasing the costs on the artificial arcs for the time periods when a product is not allowed to arrive at its destination

## 3.2 Asset management issues

We refer to the model formulation in Section 2.2 as the basic SNDAM model. We present three extensions based on ideas from real-world applications.

### 3.2.1 Fleet size and asset costs

In many service network design studies, specific fleet sizes for particular assets have to be respected. This is typically the case for problems that are closer to operational-level planning, where the degrees of freedom are fewer than in higher-level planning. By introducing a set of new constraints to the model, we are able to connect the design variables to vehicle utilization. We assume that there is an upper bound on the number of vehicles, $V_{MAX}$ , and state in (14) that $V_{MAX}$ is an upper bound on the number of operations that may be performed in each time period.

$$\sum_{(i,j)\in \mathrm{A}:T_i \le t < T_j} y_{ij} \le V_{MAX} , \qquad \forall t \in \mathrm{T} . \tag{14}$$

The fleet of vehicles is assumed to be given and the objective function in the model is not directly affected by how these vehicles are managed. It may still be useful to include an option to not utilize all vehicles in the fleet. By introducing a vehicle utilization cost, $f$ , we let the model determine the minimum number of vehicles required to perform the transportation services for the given demand. The idea is not to include aspects of investment analysis with trade-offs between fixed costs for vehicles and flow costs for products, but rather to investigate whether the fleet size could be reduced.

Vehicles (assets) operate not only when providing service, but also to move from one service to the next. Thus, introducing vehicle utilization costs require the representation of the complete vehicle workload. We present this representation in Section 3.2.3. We take this opportunity, however, to notice that, in this case, the fixed costs for opening arcs associated in the objective functions (1) and (7) to the cost of selecting/operating services, is not really required any more. Therefore, in the rest of this paper, we will replace these fixed costs with the vehicle (asset) utilization costs.

### 3.2.2 Frequency constraints

Specific lower and upper bounds may be imposed on service frequencies, i.e., on the number of occurrences of each service arc in the time-space setting. These bounds may come from operational requirements like “this service should be operated at least three times, but not more than six times a week" (Andersen et al., 2007). We label lower and upper frequencies on service arcs $(i', j') \in A'$, $L_{i'j'}$ and $U_{i'j'}$, respectively. The constraints are formulated in (15).

$$L_{i'j'} \leq \sum_{(i,j)\in A: N_i = i' \cup N_j = j'} y_{ij} \leq U_{i'j'}, \qquad \forall (i', j') \in A'. \qquad (15)$$

Several model runs would provide the means to evaluate the cost of meeting these requirements, which might be important input to decision-makers. If the costs for meeting such requirements are high, it might be worthwhile to reconsider whether these constraints are necessary for the case that is considered.

### 3.2.3 Route length constraints

In many planning problems there is the need for repetitiveness in asset rotations. Considering crew management, for example, this represents the requirements that people should leave their hometown every Monday morning. For vehicles, it may be that individual vehicles should operate the same pattern of services in all replications of the planning horizon, even if these vehicles belong to fleets of homogenous vehicles. We use Figure 2 to illustrate this idea.

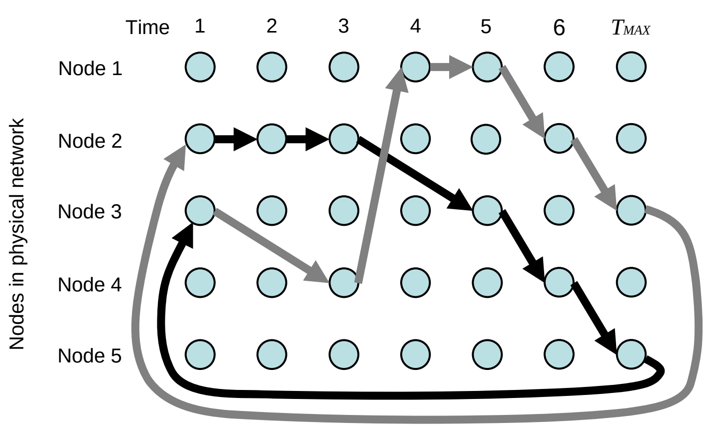



*Figure 2. Time-space network with asset routes with length equal to two planning horizons.*

In Figure 2, we have two assets that are used to open design arcs in a cyclic time-space network, and node balance constraints (13) are satisfied at all nodes. However, we observe that the assets interchange the arcs they cover at every second time period. It thus takes two repetitions of the planning horizon for the assets to return to their initial pattern. In other words, the route length for the assets corresponds to two times the length of the planning horizon. The requirement that we want to address is that these assets routes are limited to the length of the planning horizon. The operations in Figure 2 will not be feasible if we impose such route length constraints.

In order to capture the route length aspect, the model needs to have a more explicit representation of the vehicles. We introduce a set of vehicles $V = \{v\} \equiv \{1,..,V_{MAX}\}$. In addition, we need binary decision variables for each vehicle, $\delta_v$, indicating whether it is used or not. The decision variables for design arcs $y_{ij}$, have to be indexed for vehicles $v$, thus the new decision variables are $y_{ijv}$. When this change is implemented, we will have node balance constraints for each individual vehicle, not for the fleet of vehicles. The route length constraints (16) ensure that in each time period, each vehicle that is selected for operation is engaged in one activity only.

$$\sum_{(i,j)\in A:T_i \le t < T_j:} y_{ijv} - \delta_v = 0, \qquad \forall t \in T, v \in V. \qquad (16)$$

Observe that, with route length requirements, the asset movements constitute cycles in the time-space network with length equal to the length of the planning horizon, as is the case in Figure 1. It might seem that the strict cycle length of a single planning horizon imposes a rigid structure, but, as illustrated in Figure 3, the representation allows the operation of cycles shorter than this length.

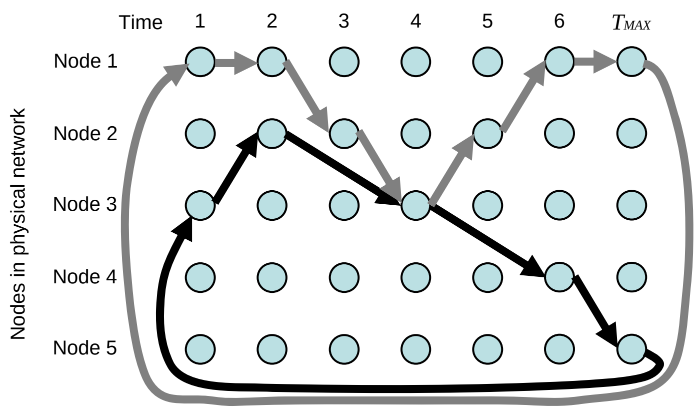



*Figure 3. Examples of asset cycles in a time-space network.*

The black arcs in Figure 3 correspond to operations that satisfy the route length constraints, the length of the resulting cycle being equal to one planning horizon. In the static service network, this time-space cycle consists of two cycles, one from node 3 at time 1 to node 3 at time 4, and one from node 3 at time 4 to node 3 at time 1. The grey arcs in Figure 3 represent another example of a feasible vehicle rotation. With three holding arcs, including the one from time $T_{MAX}$ to time 1, the operating cycle lasts for four time periods only. There is thus flexibility in the system even when route length restrictions are imposed.

## 3.3 Extended SNDAM model

Adding the extensions of Section 3.2 to the basic SNDAM formulation defined in Section 2 yields the model defined by expressions (17)-(27). The rest of this paper will be based on this formulation, which we label *extended SNDAM*.

$$Min\ z = \sum_{(i,j)\in \mathrm{A}} \sum_{p\in \mathrm{P}} c_{ij}^{p} x_{ij}^{p} + \sum_{v\in \mathrm{V}} f\,\delta_v \tag{17}$$

$$\sum_{(i,j)\in \mathrm{A}: T_i \le t < T_j :} y_{ijv} - \delta_v = 0, \qquad \forall t \in \mathrm{T}, v \in \mathrm{V}, \tag{18}$$

$$\sum_{j\in \mathrm{N}^{+}(i)} y_{ijv} - \sum_{j\in \mathrm{N}^{-}(i)} y_{jiv} = 0, \qquad \forall i \in \mathrm{N}, v \in \mathrm{V}, \tag{19}$$

$$\sum_{v\in \mathrm{V}} y_{ijv} \le 1, \qquad \forall (i,j) \in \mathrm{A}, \tag{20}$$

$$L_{i'j'} \le \sum_{\substack{(i,j)\in \mathrm{A}: \\ N_i = i' \cup N_j = j'}} y_{ijv} \le U_{i'j'}, \qquad \forall (i',j') \in \mathrm{A}', \tag{21}$$

$$\sum_{j\in \mathrm{N}^{+}(i)} x_{ij}^{p} - \sum_{j\in \mathrm{N}^{-}(i)} x_{ji}^{p} = \begin{cases} w^{p}, & i = o^{p} \\ -w^{p}, & i = d^{p} \\ 0, & otherwise \end{cases}, \qquad \forall p \in \mathrm{P}, i \in \mathrm{N}, \tag{22}$$

$$\sum_{p\in \mathrm{P}} x_{ij}^{p} - \sum_{v\in \mathrm{V}} u_{ij} y_{ijv} \le 0, \qquad \forall (i,j) \in \mathrm{A}, \tag{23}$$

$$x_{ij}^{p} - \sum_{v\in \mathrm{V}} b_{ij}^{p} y_{ijv} \le 0, \qquad \forall (i,j) \in \mathrm{A}, p \in \mathrm{P}, \tag{24}$$

$$x_{ij}^{p} \ge 0, \qquad \forall (i,j) \in \mathrm{A}, p \in \mathrm{P}, \tag{25}$$

$$y_{ijv} \in \{0,1\}, \qquad \forall (i,j) \in \mathrm{A}, v \in \mathrm{V}, \tag{26}$$

$$\delta_v \in \{0,1\}, \qquad \forall v \in \mathrm{V}. \tag{27}$$

The objective function (17) minimizes the total system cost computed as the sum of the flow costs and of the costs associated with using the assets. Constraints (18) state for each time period that if an asset is utilized, it should be engaged in one activity only. These are the route length constraints that were introduced in Section 3.2.3. Constraints (19) are the design balance constraints, while relations (20) ensure that only one vehicle can operate an arc $(i,j) \in \mathrm{A}$, which is analogous to the binary $y_{ij}$ variables in the basic SNDAM formulation (1)-(6) and (13). Lower and upper bounds on service frequencies are enforced by constraints (21), while flow balance is ensured through equations (22). Constraints (23) and (24) are the weak and strong forcing constraints, respectively, where we now aggregate over all vehicles. Finally, variable-type constraints are given in (25) to (27).

# 4 Alternate formulations of the extended SNDAM model

Solution approaches for CMND models are based on either arc- or path-based formulations referring to the decision variables associated to product flows, as discussed in Section 2.1. Obviously, a path-based formulation also exists for the basic SNDAM model. Notice that, when design balance constraints are introduced, the selection of the design arcs follows closed path-like structures. Indeed, while product paths must start and end at different nodes, namely the origin and destination nodes of the product demands, the paths of design variables must return to their starting nodes, at a different time period, to satisfy the design balance constraints. Selected design variables thus form cycles and, when route length constraints are imposed, these cycles cover all time periods exactly once. Examples of such cycles were found in Figure 3.

Four different representations of the extended SNDAM model can be written using the two different representations for the flow variables and the two representations for the design variables. To distinguish them, we introduce the following naming scheme for the SNDAM formulations:

*yx-SNDAM, where*

*y = a or c, depending of whether design variables are defined in terms of arcs or cycles*

*x = a or p, depending of whether flow variables are defined in terms of arcs or paths*

The extended model formulation (17) - (27) is therefore *aa-SNDAM*. In the following subsections we present the three other formulations to the extended SNDAM model.

## 4.1 Arc-Path formulation of SNDAM

We define paths $\mathcal{L}^p$ for product $p$ as in Section 2.1, and parameters $a_{ij}^{pl}$ =1 if arc $(i, j) \in \mathrm{A}$ belongs to path $l \in \mathcal{L}^p$ for product $p$ , 0 otherwise. The flow of product $p$ on path $l$ is $h^{pl}$, while the flow cost for transporting product $p$ on path $l$ is $k^{pl}, k^{pl} = \sum_{(i,j)\in \mathrm{A}} c_{ij}^{p} a_{ij}^{pl}$ . We can then formulate *ap-SNDAM* (arc-based design, path-based flow variables):

$$Min\ z = \sum_{p\in P}\sum_{l\in \mathcal{L}^p} k^{pl} h^{pl} + \sum_{v\in V} f\delta_v \tag{28}$$

$$\sum_{(i,j)\in A: T_i \le t < T_j:} y_{ijv} - \delta_v = 0, \qquad \forall t \in T, v \in V, \tag{29}$$

$$\sum_{j\in N^+(i)} y_{ijv} - \sum_{j\in N^-(i)} y_{jiv} = 0, \qquad \forall i \in N, v \in V, \tag{30}$$

$$\sum_{v\in V} y_{ijv} \le 1, \qquad \forall (i,j) \in A, \tag{31}$$

$$L_{i'j'} \le \sum_{\substack{(i,j)\in A:\\ N_i = i' \cup N_j = j'}} y_{ijv} \le U_{i'j'}, \qquad \forall (i',j') \in A', \tag{32}$$

$$\sum_{l\in \mathcal{L}^p} h^{pl} = w^p, \qquad \forall p \in P, \tag{33}$$

$$\sum_{p\in P}\sum_{l\in \mathcal{L}^p} a_{ij}^{pl} h^{pl} - \sum_{v\in V} u_{ij} y_{ijv} \le 0, \qquad \forall (i,j) \in A, \tag{34}$$

$$\sum_{l\in \mathcal{L}^p} a_{ij}^{pl} h^{pl} - \sum_{v\in V} b_{ij}^{p} y_{ijv} \le 0, \qquad \forall (i,j) \in A, p \in P, \tag{35}$$

$$h^{pl} \ge 0, \qquad \forall p \in P, l \in \mathcal{L}^p, \tag{36}$$

$$y_{ijv} \in \{0,1\}, \qquad \forall (i,j) \in A, v \in V, \tag{37}$$

$$\delta_v \in \{0,1\}, \qquad \forall v \in V. \tag{38}$$

The objective function (28) minimizes the sum of flow costs on paths and fixed costs for assets. Constraints (29)-(32) are identical to (19)-(22). The major change from model formulation (17)-(27) is represented by constraints (33) stating that for each product, the sum of the flows on all paths need to equal the demand. These constraints replace the flow balance constraints (22) of *aa-SNDAM*, which are ensured by the path definition. As in previous formulations, inequalities (34) and (35) are the weak and strong forcing constraints, respectively, while the variable-type constraints are presented in (36)-(38).

## 4.2 Cycle-Arc formulation of SNDAM

We introduce *cycle design* variables $g^k, \{k\} \in K$, equal to 1 if cycle $k$ is in the solution, and 0 otherwise. We also define parameter $r_{ij}^k$ equal to 1 if arc $(i,j) \in A$ is in cycle $k$, 0 otherwise. We may then formulate the *ca-SNDAM* model (cycle-based design, arc-based flow variables), where route length constraints enforce the requirement that selected cycles cover all time periods in the time-space representation. The cycles in Figure 3 are examples of feasible asset cycles. Notice that, we no longer need the asset decision variables, because each selected cycle represents a unit of asset (a vehicle). The cycle-arc version of the extended SNDAM model is presented in (39)-(47):

$$Min\ z = \sum_{(i,j)\in A} \sum_{p\in P} c_{ij}^p x_{ij}^p + \sum_{k\in K} f\, g^k \tag{39}$$

$$\sum_{k\in K} g^k \le V_{MAX}, \tag{40}$$

$$\sum_{k\in K} r_{ij}^k g^k \le 1, \qquad \forall (i,j)\in A, \tag{41}$$

$$L_{i'j'} \le \sum_{k\in K} \sum_{\substack{(i,j)\in A:\\ N_{OD_i}=i' \cup N_{OD_j}=j'}} r_{ij}^k g^k \le U_{i'j'}, \qquad \forall (i',j')\in A', \tag{42}$$

$$\sum_{j\in N^+(i)} x_{ij}^p - \sum_{j\in N^-(i)} x_{ji}^p = \begin{cases} w^p, & i=o^p \\ -w^p, & i=d^p \\ 0, & otherwise \end{cases}, \qquad \forall p\in P,\ i\in N, \tag{43}$$

$$\sum_{p\in P} x_{ij}^p - \sum_{k\in K} u_{ij} r_{ij}^k g^k \le 0, \qquad \forall (i,j)\in A, \tag{44}$$

$$x_{ij}^p - \sum_{k\in K} b_{ij}^p r_{ij}^k g^k \le 0, \qquad \forall (i,j)\in A, p\in P, \tag{45}$$

$$x_{ij}^p \ge 0, \qquad \forall (i,j)\in A, p\in P, \tag{46}$$

$$g^k \in \{0,1\}, \qquad \forall k\in K. \tag{47}$$

The objective function (39) minimizes the sum of the flow distribution costs and fixed costs for the assets that are selected and used. The structurally new constraint is (40). This constraint restricts the cycle selection, stating that the number of selected cycles is limited by the size of the available fleet of assets. This property was ensured by the dimension of set $V = \{1,..,V_{MAX}\}$ in the *aa-SNDAM* and *ap-SNDAM* formulations. Constraints (41) state that each arc $(i,j)\in A$ can be chosen by at most one cycle, which is analogous to constraints (20). In relations (42) we set the lower and upper bounds on service frequencies, while flow balance is ensured through constraints (43). Weak and strong forcing constraints are formulated in (44) and (45), respectively, while variable-type constraints are found in (46) and (47).

## 4.3 Cycle-Path formulation of SNDAM

We can combine path variables for the flow and cycle variables for the design arcs. The cycle-path formulation is presented in (48)-(56), where the objective function (48) minimizes the sum of the flow costs on paths and fixed costs for assets utilized on cycle routes. The fleet size restriction appears in (49), while selection of arc disjoint cycles is enforced by constraint (50). Lower and upper bounds on service frequencies are formulated in constraints (51), while demand satisfaction is provided by relations (52). Weak and strong forcing constraints are found in (53) and (54), respectively, while variable-type constraints appear in constraints (55) to (56).

$$Min\ z = \sum_{p \in P} \sum_{l \in \mathcal{L}^p} k^{pl} h^{pl} + \sum_{k \in K} f\, g^k \tag{48}$$

$$\sum_{k \in K} g^k \leq V_{MAX}, \tag{49}$$

$$\sum_{k \in K} r_{ij}^k g^k \leq 1, \qquad \forall (i,j) \in A, \tag{50}$$

$$L_{i'j'} \leq \sum_{k \in K} \sum_{\substack{(i,j) \in A: \\ N_{OD_i} = i' \cup N_{OD_j} = j'}} r_{ij}^k g^k \leq U_{i'j'}, \qquad \forall (i',j') \in A', \tag{51}$$

$$\sum_{l \in \mathcal{L}^p} h^{pl} = w^p, \qquad \forall p \in P, \tag{52}$$

$$\sum_{p \in P} \sum_{l \in \mathcal{L}^p} a_{ij}^{pl} h^{pl} - \sum_{k \in K} u_{ij} r_{ij}^k g^k \leq 0, \qquad \forall (i,j) \in A, \tag{53}$$

$$\sum_{l \in \mathcal{L}^p} a_{ij}^{pl} h^{pl} - \sum_{k \in K} b_{ij}^p r_{ij}^k g^k \leq 0, \qquad \forall (i,j) \in A, p \in P, \tag{54}$$

$$h^{pl} \geq 0, \qquad \forall p \in P, l \in \mathcal{L}^p, \tag{55}$$

$$g^k \in \{0,1\}, \qquad \forall k \in K. \tag{56}$$

# 5 Algorithms for path and cycle generation

CMND models may be solved in a variety of ways, see for instance Magnanti and Wong (1984) and Gendron et al. (1999). The development of particular methods for the models proposed is well beyond the scope of this paper. Our goal is rather to compare the different formulations of the extended SNDAM model in order to identify their strengths and weaknesses with respect to computational efficiency, and to study the impact of asset management considerations. We perform this comparison on problem instances where all paths and cycles have been generated *a priori*. This avoids the unnecessary burden of developing progressive variable generation-based procedures that, together with meta-heuristics, are required to address larger instances and should be the scope of future work. We describe algorithms for generation of flow paths and design cycles in Sections 5.1 and 5.2, respectively.

## 5.1 Path generation

We now present an algorithm for complete a priori generation of flow paths. We assume that there are no connections between nodes of the same time period and that the path length cannot exceed the length of the planning horizon. The algorithm considers all outgoing arcs from the origin node of a product. For each of these arcs, a path, represented as a linked list is created. These paths constitute an initial pool of candidate paths. The algorithm then traverses all nodes, starting from the closest successor among the nodes that are reached by the outgoing arcs from the origin node. For all candidate paths whose last node so far is the current node, the candidate paths are updated with the outgoing arcs from these nodes. When a node belonging to the product physical destination node is reached, a feasible product path from origin to destination is found. A dominance rule is introduced to avoid an exponentially growing number of paths as a consequence of the existence of holding arcs: If a candidate path has already visited a physical node, it should not return to the same physical node later, since using the uncapacitated holding arcs in this case is at least as cheap as using movement arcs. The algorithm is presented in Figure 4.

```
forall(p in Product) do
   forall(arcs starting in the product’s origin) do
        create candidate list based on the arc;
   end-do

   define candidate nodes to be nodes from the time period after the product becomes
       available to the end of the planning horizon, and again starting from the first
       period until two time periods before the product becomes available;

   forall(candidate nodes) do
     forall(candidate lists where last node = this node) do
       forall(emanating arcs of the candidate node) do
         if(destination node of arc belongs to the product’s destination node) then
            a path has been found, store it;
         else
             if(visited=false for the appurtenant physical node) then
                 if(destination of arc is reached without crossing the time period where
                    the product becomes available) then
                        add arc to candidate path;
                        visited of appurtenant physical node of candidate path=true;
                 end-if
             else
                 if(the origin and destination of the arc have the same appurtenant
                    physical node) then
                        add arc to candidate path;
                 end-if
             end-if
          end-if
        end-do
      end-do
   end-do
end-do
```

*Figure 4. Pseudo-code for the path generation algorithm.*

## 5.2 Cycle generation

A cycle is a set of consecutive arcs in the time-space network, a path, with length equal to the length of the planning horizon, and closed such that one will return to the initial node after traversing all the arcs. To build cycles, we start with all arcs emanating from nodes in the first time period, and define each of these arcs to be the first arc in a candidate path. Then, for nodes in time period 2 and later, we check if the candidate paths have this node as their current last node. In the affirmative, the outgoing arcs of this node are added to these candidate paths. Thus, for example, if there are three emanating arcs at a node, each candidate path that has this node as its current last node yields three candidate paths, one for each emanating arc. If the destination node of such an outgoing arc is the origin node of the candidate path, a cycle has been found. However, if the destination node of the arc is such that the length of the candidate path is longer than or equal to the length of the planning horizon and no cycle has been established, then this candidate path is not updated with the arc, because it cannot result in a feasible cycle covering all time periods in the planning horizon exactly once.

When all arcs have duration of one time period, this approach is sufficient to create all possible cycles. However, when arcs with longer durations exist, paths that do not include nodes at period 1 may be overlooked. To avoid such a case, we include a second phase where candidate paths are established from all arcs that are cross-horizon arcs (origin node has a smaller node number than the destination node) and end *after* time period 1.

The algorithm for cycle generation thus consists of three phases, where the two first phases generate initial candidate paths and the third phase traverses the network and searches for cycles. In Figure 5 we illustrate one cycle, solid lined, which would be created from the first phase of the algorithm, and a dotted cycle that would be created from the second phase of the algorithm.

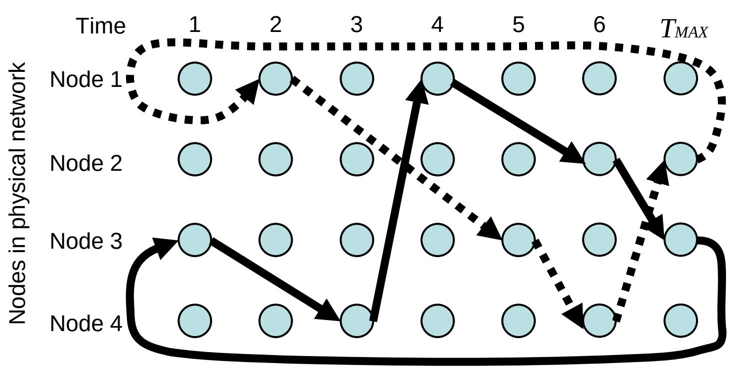



*Figure 5. Cycles that are created from the first (solid) and second (dotted) phase of the algorithm.*

There are no restrictions on asset movement during the cycle generation. Thus, for instance, a vehicle may enter and leave a terminal (node) several times throughout a weekly planning horizon. The algorithm for cycle generation is summarized in Figure 6.

# 6 Computational study

The purpose of the computational study is to evaluate the performance of the different formulations of the extended SNDAM model with respect to solution times - and to study the effect of asset management on solution times and on the plans obtained from the models. For cycles and paths, we use the a priori generation procedures presented in Section 5. We are interested, in particular, whether the introduction of cycles seems useful and, if this is the case, under what circumstances do the associated formulations perform well.

To explore these issues, we create a set of problem instances with varying dimensions. On the one hand, we create instances with 15, 20, and 25 time periods, and 15, 20, and 25 products, producing 9 scenarios. On the other hand, we explore two different network densities, representing sparse and dense networks, building networks with 5 physical nodes and 10 and 15 arcs in each time period, respectively. Finally, we generate three 15 arcs and 15 time periods scenarios where we increased the number of products to 50, 100 and 200. The 21 data sets and corresponding scenarios are labeled *nwaxtypz*, where *w* is the number of physical nodes, *x* is the number of physical arcs excluding holding arcs, *y* is the number of time periods, and *z* is the number of products.

We work with the strong formulations of the problem throughout the computational study, implying that the strong forcing constraints are included in the formulations. This increases the size of the problems considerably, which again reduces the tractable problem dimensions. Yet, because for general CMND models the strength of the linear relaxations is improved significantly when strong forcing constraints are considered, we find it relevant to consider the strong formulations for the SNDAM as well.

```
Initialization phase 1
forall(nodes in first time period) do
  forall(emanating arcs) do
    create candidate path based on the arc;
  end-do
end-do
Initialization phase 2
forall(nodes after first time period) do
  forall(emanating arcs) do
    if(the arc is a cross-horizon arc and ends after the first period) then
      create candidate path based on the arc;
    end-if
  end-do
end-do
Joint cycle generation phase
forall(nodes after first time period) do
  forall(candidate paths where last node = this node) do
    forall(emanating arcs of this node) do
      if(destination of arc = the paths origin) then
         a cycle has been found, store it;
      else
         if(destination of arc > the paths origin) then
           skip arc because we cannot make a cycle;
         else
           add arc to candidate path;
         end-if
      end-if
    end-do
  end-do
end-do
```

*Figure 6. Pseudo-code for the cycle generation algorithm.*

We compare the four SNDAM formulations in Section 6.1 using the 21 scenarios defined above. To make a stronger claim on the representativeness of the results, we also compare the formulations on five additional data sets generated for a subset of the scenarios. Section 6.2 is dedicated to the study of the impact of the asset management issues. We choose a subset of the scenarios from Section 6.1 and examine how solution times and optimal integer solution values are affected by the various asset management issues of the extended SNDAM model.

The solution times of the arc-based formulations are sensitive to the fleet size, because a large number of vehicles introduces symmetry and increases the number of binary design variables. To mitigate this impact, we always solve the cycle-based formulations first, and use the resulting optimal number of vehicles as the asset fleet size when solving the arc-based formulations.

All formulations are modeled in XPRESS-IVE version 1.17.04 and matrix files are created on a computer with an Intel Dual Core 2,3Ghz processor and 2GB memory. The matrix files are imported to CPLEX 10.0.0 and solved using a 2.4 GHz AMD Opteron 250 processor with 16GB memory. The algorithms for a priori generation of flow paths and design cycles are implemented in Visual C++.

## 6.1 Comparing the SNDAM formulations

Table 1 displays solution times for finding the optimal integer solution for the four formulations of the extended SNDAM problem in columns 1 to 4, respectively. All problems were run for 10 hours (36 000 CPU seconds). “No integer” is marked when no integer solution was found within 10 hours of CPU time. The gap between the lower and upper bounds at termination is reported when integer solutions were found, but optimality could not be proven within 10 hours of CPU time.

The last two columns of Table 1 display the gaps between the strong linear relaxations at the *root node* of the branch-and-bound with the two arc- and the two cycle-based formulations, respectively, and the best integer solution obtained within 10 hours of CPU time with any of the four formulations. For most of the instances, this integer solution was the optimal integer solution of the problem. These gaps gave an indication of the tightness of the strong linear relaxations of the formulations.

We observe from Table 1 that the cycle-based formulations perform very well compared to the formulations with design arcs. In all scenarios, the fastest solution time was obtained with one of the cycle-based formulations. It is not so that both cycle-based formulations are the best for all the problems but, in general, they outperform the arc-based formulations. One interesting observation is that for given numbers of physical arcs and products, increases in the number of time periods increases the solution times for the cycle-based formulations. The reason for this is that the number of cycles grows exponentially with the number of time periods. For arc-based formulations, however, we observe the opposite effect in several cases; the solution times are reduced with an increased number of time periods. To explain this, we observe that for a given demand and duration of the arcs, one vehicle can operate more services during the planning horizon with more time periods, and fewer vehicles are thus needed to cover the demand. The latter reduces the problem with symmetry in the arc-based formulations, which are based on vehicle-indexed design arcs. However, if we increase the number of products for a given number of time periods, the solution time increases also for the arc-arc and arc-path formulations. To summarize the discussion, cycle-based formulations appear to perform significantly better than the formulations with design arcs, and the difference in solution times increases if the ratio between volumes shipped and number of time periods increases.

*Table 1. Solution times (CPU seconds) and gaps between strong linear relaxation at root node and best integer solution. Optimality gap is reported if solution time exceeds 36 000 sec.*

| Problem | Arc-arc solution time | Arc-path solution time | Cycle-arc solution time | Cycle-path solution time | Strong LP-gap aa and ap | Strong LP-gap ca and cp |
|---|---|---|---|---|---|---|
| n5a10t15p15 | 11 160 | 8 611 | 6 | 7 | 22 % | 9 % |
| n5a10t20p15 | 6 396 | 3 322 | 32 | 86 | 13 % | 6 % |
| n5a10t25p15 | 204 | 264 | 52 | 77 | 5 % | 5 % |
| n5a10t15p20 | 17 899 | 19 555 | 19 | 14 | 12 % | 3 % |
| n5a10t20p20 | 3 658 | 3 481 | 17 | 9 | 6 % | 4 % |
| n5a10t25p20 | 644 | 942 | 224 | 3 805 | 4 % | 4 % |
| n5a10t15p25 | gap 3.6% | gap 4.5% | 4 | 3 | 13 % | 4 % |
| n5a10t20p25 | gap 6.0% | gap 3.9% | 7 | 3 | 12 % | 2 % |
| n5a10t25p25 | 942 | 2 724 | 251 | 398 | 4 % | 3 % |
| n5a15t15p15 | 1 159 | 245 | <0.5 | <0.5 | 9 % | 0.5 % |
| n5a15t20p15 | 2 871 | 3 984 | 25 | 9 | 9 % | 6 % |
| n5a15t25p15 | 348 | 139 | 56 | 35 | 4 % | 0.0 % |
| n5a15t15p20 | 2 311 | 1 202 | 1 | 1 | 6 % | 1 % |
| n5a15t20p20 | 7 327 | 5 375 | 19 | 25 | 4 % | 1 % |
| n5a15t25p20 | gap 1.5% | gap 3.0% | 367 | 354 | 6 % | 2 % |
| n5a15t15p25 | gap 4.6% | gap 4.8% | 3 | 3 | 11 % | 5 % |
| n5a15t20p25 | gap 4.1% | gap 4.1% | 1 263 | 1 235 | 7 % | 4 % |
| n5a15t25p25 | 26 258 | 8 479 | 213 | 143 | 4 % | 0.2 % |
| n5a15t15p50 | gap 10.9% | gap 8.4% | 8 406 | 2 762 | 10 % | 2 % |
| n5a15t15p100 | No integer | gap 10.4% | gap 0.8% | gap 0.7% | 9 % | 2 % |
| n5a15t15p200 | No integer | No integer | gap 0.6% | gap 0.5% | 20 % | 1 % |

Capacitated multicommodity fixed charge network design problems usually have poor linear relaxations (Gendron and Crainic, 1996), which is one important reason for the difficulty associated with addressing CMND models. In the last two columns of Table 1, we report the gap between the strong linear relaxation at the root node of the branch-and-bound tree with arc- and cycle-based formulations, and the best integer solution that was obtained for the specific instance with any of the four formulations. Table 1 indicates that there is a significant difference between the formulations with arc-based and cycle-based design variables when it comes to the strength of the linear relaxation. While the arc-arc and the arc-path formulations typically yield gaps from 5% to 20%, the cycle-arc and cycle-path formulations display gaps from 1% to 5%. This weakness of the lower bounds contributes certainly to the long solution times of the arc-arc and the arc-path formulation reported in Table 1. The results of the computational study thus indicate that the introduction of cycle-based design variables constitutes a promising research direction for efficient solution methods for service network design problems with asset management.

For the last three scenarios reported in Table 1, the number of products is increased towards more realistic values. We observe that the cycle-based formulations become more difficult to solve when the dimensions of the problems increase; optimal solution cannot be proved for the 100 and 200 product problem instances within 10 hours of CPU time. However, the cycle-based formulations still outperform arc-based formulations. For 200 products, none of the arc-based formulations produce integer solutions, while for 100 products, only one integer solution is found for the arc-path formulation, with an optimality gap of 10.4% after 10 hours of CPU time. We thus conclude that cycle-based formulations offer the most attractive perspective for designing solution methods for this class of problems.

To validate the results and strengthen the conclusions, we compared the four formulations on five additional data sets for six of the dimensions reported in Table 1. Table 2 displays the results. Scores are used to compare the formulations. A score of 3, 2, 1 or 0 points is given, where 3 points stands for the best performance, and 0 points for the worst performance. The computational time is limited to 3 hours (10 800 seconds), because the results of the first experiment reported in Table 1 indicate that if the optimal solution is found, it is usually found within reasonable time. We report the scores, as well as the number of problems that were not solved to optimality within 3 hours of CPU time.

The results reported in Table 2 support the conclusions of the experiment reported in Table 1: Cycle-based formulations appear to be more easily solved than arc-based formulations. The cycle-path formulation appears to perform best, and it is only for this formulation that all problems are solved to optimality within 3 hours of CPU time. With the arc-arc and arc-path formulations, we are only able to solve to optimality 8 and 10 out of the 30 scenarios, respectively.

*Table 2. Scored performance of the four formulations on the additional data sets. Maximum CPU time is 3 hours (10 800 sec).*

| Scenario | Performance score | | | | Number of instances solved to optimality | | | |
|---|---|---|---|---|---|---|---|---|
| | Arc-Arc | Arc-Path | Cycle-Arc | Cycle-Path | Arc-Arc | Arc-Path | Cycle-Arc | Cycle-Path |
| n5a10t25p25 | 0 | 5 | 11 | 14 | 0 | 1 | 5 | 5 |
| n5a15t15p15 | 2 | 3 | 10 | 15 | 3 | 3 | 5 | 5 |
| n5a15t20p15 | 3 | 2 | 12 | 13 | 4 | 4 | 5 | 5 |
| n5a15t15p25 | 2 | 3 | 10 | 15 | 0 | 0 | 5 | 5 |
| n5a15t20p25 | 4 | 1 | 12 | 13 | 1 | 2 | 5 | 5 |
| n5a15t15p100 | 2 | 3 | 10 | 13 | 0 | 0 | 3 | 5 |
| Total | 13 | 17 | 65 | 83 | 8 | 10 | 28 | 30 |

## 6.2 Effects of asset management issues

In this subsection, we study the impact of the asset management considerations discussed in Section 3 on the solutions obtained by the optimization models. To perform the analysis, we remove groups of constraints from the original extended SNDAM formulation, and observe changes in solution times and optimal integer solutions. All computations are based on the arc-arc formulation, because cycle-based formulations are not relevant when the repetitiveness and design balance constraints are left out of the model. Because we gradually remove the concept of assets from the model formulation, we have to change the representation of costs. Instead of having asset costs, we reintroduce costs on design arcs as in Section 2, with costs such that a set of design arcs covering the planning horizon exactly once in total have costs corresponding to one entity of asset costs. Table 3 summarizes the results of six tests on five scenarios from Section 6.1. Each test represents a change in the asset management issues considered. When constraints are removed from the original model, we expect the optimal integer solution to either improve or remain unchanged. Improvements in the objective function appear as (relative) reductions in Table 3 because we have a minimization problem.

*Table 3. Effects of removing asset management constraints. Changes in % in objective value and time for finding optimal integer solution relative to the full asset management models in Section 6.1.*

| | Route length | | Route length and design balance | | Route length and fleet size | | Route length, design balance and fleet size | | Bounds on service frequencies | | All asset management constraints | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Scenario | Obj. value | Sol. time | Obj. value | Sol. time | Obj. value | Sol. time | Obj. value | Sol. time | Obj. value | Sol. time | Obj. value | Sol. time |
| n5a10t25p15 | 0 % | -79 % | -5 % | -23 % | 0 % | -60 % | -17 % | -96 % | 0 % | 7 % | -17 % | -96 % |
| n5a15t15p25 | -7 % | -100 % | -11 % | -92 % | -7 % | -100 % | -17 % | -100 % | -7 % | -65 % | -18 % | -100 % |
| n5a15t20p20 | 0 % | -77 % | -9 % | -94 % | 0 % | -72 % | -15 % | -99 % | -2 % | -65 % | -16 % | -99 % |
| n5a15t20p25 | -5 % | -100 % | -13 % | -100 % | -5 % | -99 % | -17 % | -100 % | -6 % | 0 % | -25 % | -100 % |
| n5a15t25p20 | -1 % | -100 % | -13 % | -100 % | -1 % | -99 % | -16 % | -100 % | -8 % | -36 % | -25 % | -100 % |
| Average | -3 % | -91 % | -10 % | -82 % | -3 % | -86 % | -17 % | -99 % | -5 % | -32 % | -20 % | -99 % |

We observe that removal of the route length constraints has a minor impact on the values of the integer solutions. When these constraints are removed, the optimal objective function value is unchanged in two scenarios, while only minor improvements are observed for the other scenarios. Solution times are strongly reduced, however, by 91% on average. This is partly due to the removal of the vehicle indexation of the design arcs. Reversing the perspective, we observe that we may add route length constraints to the basic SNDAM model without significant degradation in objective function value, but with a cost in computational performance.

In the next two tests, we remove design balance and fleet size constraints, respectively, *in addition to* the route length constraints. The additional removal of fleet size constraints did not affect the optimal objective value for these problems. However, removal of design balance constraints had significant influence on the objective value, with a total reduction of 10% on average. In the fourth test, removal of both design balance and fleet size constraints in addition to the route length constraints, yields on average a 17% improvement in objective function values. It thus appears that, fleet size constraints influence the solution only when design balance constraints are not included. This finding is reasonable, because as long as design balance constraints are there, opening an "incredibly good" service arc implicitly implies that several other (not so good) design arcs have to be opened, and the cost increases. But, when design balance constraints are removed, we are free to open a range of good service arcs without further obligations, and because of that, removing fleet size constraints might make it advantageous to have more simultaneous operations than the fleet size allowed for. We should remark, however, that for the test problems studied in this paper, we had a special adaptation of setting the fleet size as low as possible to reduce the problem with symmetry in the arc-based formulations.

The fifth test, column 6 in Table 3, considered the impact of bounds on service frequencies. The objective function value improved by 5% on average. We observe that, in this case, solution times are not as strongly reduced as in the other tests, and even increasing for one of the instances. Route length constraints were not removed in this test, and these constraints appear to be the far most significant cause of the reduced solution times in all the other tests reported in Table 1.

Finally, we removed all asset management constraints and, as reported in the last column of Table 3, we observed dramatic improvements in computational performance together with a decrease in objective function value in the 16% to 25% range.

To conclude, asset management issues impact the design of the services and the efficiency of the solution procedure of the standard MIP-solver of CPLEX. The most significant impact on the solution times was found by removing the route length constraints. Moreover, it appears as if the design balance constraints in particular have an impact on the objective function values that are obtained. The latter is not a surprising result, as these constraints considerably affect the flexibility in the design decisions.

# 7 Concluding remarks

In this paper, we have studied the impact of considering asset management in service network design. Asset management reflects important issues connected to vehicle management and utilization in real-world design studies. We have proposed an extended model for Service Network Design with Asset Management (SNDAM), introduced several formulations for the model, and performed an extensive set of analyzes both on the relative performance of the formulations and the impact of asset management-based constraints on the service network design problem.

Based on a cyclic time-space representation capturing the scheduling aspect of real-world transportations services, we defined cycle design variables. A cycle represents a feasible asset (vehicle) route through the planning horizon supporting selected services. It is thus a joint selection of design arcs that together cover the planning horizon exactly once. In addition, we introduced path variables for flows as an alternative to arc variables. Combining the two representations of the design variables and the two representations of the flow variables, we introduced four alternative formulations of the extended SNDAM problem.

The computational study compared model solving based on the four formulations on problem instances for which paths and cycles were generated a priori. The computational study showed that the formulations of SNDAM based on design cycle variables may be solved significantly faster than the formulations based on design arc variables. One interesting observation in this respect is that the cycle-based formulations display a significantly stronger *strong linear relaxation* than arc-based formulations. This is encouraging for the development of efficient solution methods, because poor lower bounding is one of the challenges in traditional network design (Gendron and Crainic, 1996).

We also analyzed the effect of the considered asset management issues on the efficiency of model solving and on the values of the integer solutions that were obtained by solving the models. The overall conclusion is that these issues indeed affect the optimal solutions obtained and that, in general, they increase the computational effort required to solve the extended formulations. This emphasizes the need to focus future research on developing efficient solution approaches for real-world-dimensioned service network design problems that include aspects of asset management.

The test problems that we use in this paper are smaller than many instances from real-world planning problems. The number of cycles that have to be considered in a full enumeration scheme increases considerably when the number of time periods increases for a given physical network. As a rule of thumb, we observed that increasing the number of time periods by five increased the number of cycles by a factor of ten. We have tested an alternative cycle representation where holding arcs are removed and waiting instead is introduced as part of movement arcs. With this approach we were able to solve slightly larger instances than with the approach used in this paper, but no *significant* improvements in tractability were found. Different solution methods are thus required in order to handle larger instances. The definition of cycle variables naturally suggests dynamic column generation as part of these methods. The combinatorial nature of the problem and previous work on network design hints that meta-heuristics and parallel computing will probably also be part of the solution. The algorithms that we proposed for the a priori path and cycle generation contain ideas that may be explored in this context. We are currently working on these topics and expect to report results in the near future.

# Acknowledgements

This work has received financial support from The Norwegian Research Council through the Polcorridor Logchain project. Partial funding has also been supplied by the Natural Sciences and Engineering Research Council of Canada (NSERC) through its Discovery and Industrial Research Chair grant programs.

# References


Ahuja, R.K., Magnanti, T.L., Orlin, J.B., 1993. *Network Flows*. Prentice Hall, Upper Saddle River, NJ.

Andersen, J., Crainic, T.G., Christiansen, M., 2007. Service network design with management and coordination of multiple fleets. *European Journal of Operational Research*.

Barnhart, C., Schneur, R.R, 1996. Air network design for express shipment service. *Operations Research* 44 (6), 852- 863.

Crainic, T.G., 2000. Service network design in freight transportation. *European Journal of Operational Research* 122(2), 272-288.

Crainic, T.G., Laporte, G., 1997. Planning models for freight transportation. *European Journal of Operational Research* 97, 409-438.

Crainic, T.G., Gendreau, M., Farvolden, J.M., 2000. A Simplex-based Tabu Search for Capacitated Network Design. *INFORMS Journal on Computing* 12(3), 223-236.

Crainic, T.G., Frangioni, A., Gendron, B., 2001. Bundle-based relaxation methods for multicommodity capacitated fixed charge network design. *Discrete Applied Mathematics* 112, 73-99.

Gendron, B., Crainic, T.G., 1996. *Bounding procedures for multicommodity capacitated fixed charge network design problems*. Publication CRT-96-06, Centre de recherche sur les transports, Université de Montréal, Quebec, Canada.

Gendron, B., Crainic, T.G., Frangioni, A., 1999. Multicommodity capacitated network design. In B. Sansò, P. Soriano (Eds.), *Telecommunications Network Planning*, Kluwer Academic Publishers, Doordrecht, 1999, pp. 1–19 (Chapter 1).

Kim, D., Barnhart, C., Ware, K., Reinhardt, G., 1999. Multimodal express package delivery: A service network design application. *Transportation Science* 33(4), 391-407.

Lai, M.F., Lo, H.K., 2004. Ferry service network design: optimal fleet size, routing and scheduling. *Transportation Research Part A: Policy and Practice* 38, 305-328.

Macharis, C., Bontekoning, Y.M., 2004. Opportunities for OR in intermodal freight transport research: A review. *European Journal of Operational Research* 153, 400-416.

Magnanti, T.L., Wong, R.T., 1984. Network design and transportation planning: models and algorithms. *Transportation Science* 18(1), 1-55.

Pedersen, M.B., 2006. *Optimization models and solution methods for intermodal transportation*. PhD thesis, Centre for Traffic and Transport, Technical University of Denmark.

Pedersen, M.B., Crainic, T.G., 2007. *Optimization of intermodal freight train service schedules on train canals*. Publication, Centre de recherche sur les transports, Université de Montréal.

Pedersen, M.B., Crainic, T.G., Madsen, O.B.G., 2007 *Models and tabu search meta-heuristics for service network design with asset-balance requirements*. Publication, Centre de recherche sur les transports, Université de Montréal.

Smilowitz, K.R., Atamtürk, A., Daganzo, C.F., 2003. Deferred item and vehicle routing within integrated networks. *Transportation Research Part E: Logistics and Transportation* 39, 305-323.