

Service and Transfer Selection for Freights in a Synchromodal Network

Arturo Pérez Rivera^(✉) and Martijn Mes

Department of Industrial Engineering and Business Information Systems,
University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands
{a.e.perezrivera,m.r.k.mes}@utwente.nl

Abstract. We study the problem of selecting services and transfers in a synchromodal network to transport freights with different characteristics, over a multi-period horizon. The evolution of the network over time is determined by the decisions made, the schedule of the services, and the new freights that arrive each period. Although freights become known gradually over time, the planner has probabilistic knowledge about their arrival. Using this knowledge, the planner balances current and future costs at each period, with the objective of minimizing the expected costs over the entire horizon. To model this stochastic finite horizon optimization problem, we propose a Markov Decision Process (MDP) model. To overcome the computational complexity of solving the MDP, we propose a heuristic approach based on approximate dynamic programming. Using different problem settings, we show that our look-ahead approach has significant benefits compared to a benchmark heuristic.

Keywords: Synchromodal planning · Anticipatory planning · Intermodal transport · Approximate dynamic programming

1 Introduction

We consider the problem of selecting services and transfers in a synchromodal network, to transport freights from their origin to their destination, while minimizing costs over a multi-period horizon. In a synchromodal setting, all freights are booked “mode-free”, meaning that there are no restrictions for selecting a transportation mode or deciding the number transfers among the intermodal terminals. The network operator is able to decide over all services in the network even if they are not its own. However, the flexibility in selecting services and transfers is encumbered by various time restrictions, such as service schedules and freight time-windows, and by the variability in the arrival of freights over time. In this paper, we study how these challenges can be tackled, heuristically, in order to solve this stochastic and finite horizon optimization problem.

In synchromodal planning, it is possible to change the transportation plan to bring a freight from its origin to its destination, at any point in time. Even though the planner might have a complete plan at a given moment, only the first part

of such a plan is implemented. The next decision moment, the planner has the flexibility to change the original plan if necessary. Consequently, there are three types of decisions each day: (i) transport a freight to its final destination, (ii) transport a freight to an intermediate terminal, and (iii) postpone the transport of a freight. All types of decisions incur some form of costs. The first and the second type incur *direct* costs, which are costs realized by the services required for the transportation of a freight. The third type has direct costs only in case of holding costs. Since the problem is to minimize costs over a multi-period horizon, the second and third type also incur *future* costs, which are costs that are incurred on a posterior moment within the horizon. Future costs depend on the new arriving freights and their characteristics (which are uncertain) and the known transportation mode characteristics (e.g., schedules, capacity, etc.). The optimal balance between direct and future costs guarantees the best performance over the horizon. However, anticipating future costs is challenging.

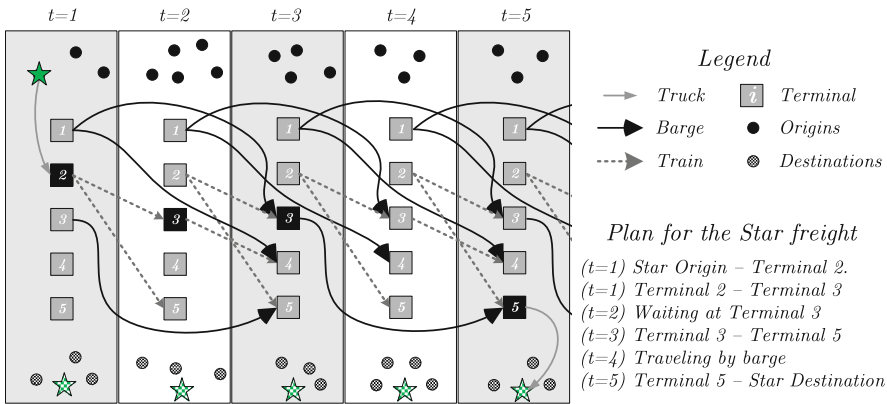


Fig. 1. Time evolution and planning example of service and transfer selection

Decisions are influenced by two types of time restrictions. The first type corresponds to the durations and schedules of services and transfers. As an example, consider Fig. 1, which shows a possible plan spanning 5 days using both train and barge. In this example, barges have a duration of 2 days, and the train between Terminals 3 and 4 departs on even days. The second type corresponds to the time-window of freights, which limit the feasible transportation services and transfers, and thus the feasible decisions. In addition to the time restrictions, the variability in the number of freights that arrive each day and their characteristics (i.e., origin, destination, time-window), also influence the decisions. Although freights and their characteristics are unknown beforehand, there is probabilistic information about their arrival. Every day, the planner must consider all these characteristics and select which freights use the services available that day, balancing direct and future costs.

The objective of this paper is twofold: (i) to design a model and look-ahead solution method that capture all problem characteristics and their effect on the planning objective, and (ii) to explore the use of look-ahead decision methods under several settings. We model the decision problem and the evolution of the network using a Markov Decision Process (MDP) model. With this model, the optimal trade-off between the three types of decisions, over time and under uncertain demand, can be obtained. However, solving MDP models become unmanageable as problem instances grow larger. To overcome this, we use Approximate Dynamic Programming (ADP), a framework that uses parts from the MDP model and iteratively estimates future costs. ADP combines simulation, optimization, and statistical techniques to solve an MDP heuristically.

The remainder of this paper is organized as follows. In Sect. 2, we briefly mention the relevant literature and specify our contribution to it. In Sect. 3, we introduce the MDP model. In Sect. 4, we explain our ADP solution approach. In Sect. 5, we test various designs within the ADP algorithm, and provide a comparison with benchmark heuristics. Finally, we close in Sect. 6 with conclusions and insights for further research.

2 Literature Review

In this section, we focus our attention on the literature about planning problems in dynamic and flexible intermodal transportation networks. It is our goal to provide an overview of the advantages and limitations of related work, i.e. possible solution methods. Extensive literature reviews about this area and thorough explanation of modeling and solution approaches can be found in [2, 14].

Sychromodal planning is the proactive organization and control of intermodal transportation services based on the latest information available [14]. In such a planning paradigm, decision methods must balance the demand with all available services and intermodal transfers each time new information becomes known [13]. Although research about sychromodal planning methods is on its infancy, several studies show how existing methods for intermodal transport planning can be extended to such problem settings [16] and how significant gains can be achieved in practice [9, 16].

In intermodal transport planning, Dynamic Service Network Design (DSND) problems are the closest to the sychromodal planning problems. DSND involves the selection of transportation services and modes for freights, where at least one feature of the network varies over time [14]. Due to the time-space nature of DSND problems, graph theory and mathematical programming approaches are commonly used in this area. However, these approaches have computational limitations for large and complex time-evolving problem instances [15], which are characteristics common to sychromodality [13]. To overcome these limitations, additional designs, such as decomposition algorithms [5], receding horizons [6], and model predictive control [10], are necessary. These additional designs are less suitable for including probabilistic information in the decisions, which may explain why most DSND studies assume deterministic demand [14] even though the need to incorporate stochastic demand has been recognized [7].

To incorporate stochasticity in DSND approaches, techniques such as scenario generation [3, 7], two-stage stochastic programming [1, 8], and Approximate Dynamic Programming (ADP) [4, 11] have been used. Although these approaches perform better than their deterministic counterpart, they have limitations when considering synchromodal planning. In the scenario generation technique, plans do not change as new information becomes available. In the two-stage stochastic programming approach, explicit probabilistic constraints and high computational requirements limit their applicability to large instances. In ADP, a proper design and validation of the approximation algorithm is crucial and challenging. Nevertheless, ADP allows generic modeling of complex, time-revealing, stochastic networks and a fast response time for updating plans.

To summarize, DSND research provides a useful base for synchromodal planning. Considering all challenges and opportunities mentioned before, we believe that our contribution to the literature about stochastic DSND problems and synchromodal planning has three key points. First, we design an MDP model and solution method based on ADP that capture all problem characteristics and their effect on the planning objective. Second, we explore the use of such a look-ahead approach, under different problem settings, and provide design and validation insights. Third, we compare the ADP approach against an advanced sampling procedure and specify further research directions based on the insights.

3 Optimization Model

In this section, we present our optimization model. Following the DSND convention, we begin presenting the network parameters using a directed graph. Then, we present the MDP model for our stochastic planning problem.

3.1 Input Parameters

We define a directed graph $G_t = (\mathcal{N}_t, \mathcal{A}_t)$, where $t \in \mathcal{T} = \{0, 1, 2, \dots, T^{\max} - 1\}$ represents the finite planning horizon (i.e., T^{\max} decision periods), \mathcal{N}_t represents the set of all nodes at time t , and \mathcal{A}_t represents the set of all directed arcs at time t . In the remainder of the paper, we refer to a time period t as a day, although it is important to note that time can be discretized in any arbitrary interval. Also in the remainder of the model description, all notation and formulations indexed by t correspond to that day. Nodes \mathcal{N}_t represent physical locations where freight can begin or end transportation, i.e., origins, intermodal terminals, and destinations. We denote the set of origin nodes as \mathcal{N}_t^O , the set of destination nodes as \mathcal{N}_t^D , and the set of intermodal terminal nodes as \mathcal{N}_t^I . These three sets are mutually exclusive, make up the set of all nodes, and are all indexed with i, j and d . Note that this separation of the node sets applies to the model, but not necessarily to the problem instance. For example, an intermodal terminal which receives arriving freights will be modeled both as an origin and terminal node, with all services between these two nodes properly adjusted. Arcs \mathcal{A}_t represent all transportation services in the network. Similar to the node classification,

we classify the arcs into three types. The set of arcs between an origin and an intermodal node is denoted as $\mathcal{A}_t^O = \{(i, j) | i \in \mathcal{N}_t^O \text{ and } j \in \mathcal{N}_t^I\}$. The set of arcs between two intermodal terminal nodes is denoted as $\mathcal{A}_t^I = \{(i, j) | i, j \in \mathcal{N}_t^I\}$. The set of arcs between an origin or an intermodal node, and a destination, is denoted as $\mathcal{A}_t^D = \{(i, d) | i \in \mathcal{N}_t^O \cup \mathcal{N}_t^I \text{ and } d \in \mathcal{N}_t^D\}$.

We make three modeling assumptions with respect to the services between different types of locations. First, we assume that services beginning at an origin, i.e., \mathcal{A}_t^O , as well as services ending in a destination, i.e., \mathcal{A}_t^D , are available every day and are realized by truck. This assumption corresponds to the usual pre- and end-haulage operations in an sychromodal network. Second, we assume that services between two intermodal terminals, i.e., \mathcal{A}_t^I , are done by high-capacity modes and never by truck. Although this is a simplification of the network, trucks between intermodal terminals are rarely used. If the problem instance requires it, a truck service between two intermodal terminals can be modeled using “dummy” nodes for the respective terminals, with other arcs properly adjusted. Third, we assume there is at most one service between two intermodal terminal nodes. Just as before, multiple services between two intermodal terminals can be modeled using more than one pair of nodes representing those terminals. Note that the services between two intermodal terminals are not necessarily the same every day to represent the schedules for the high-capacity modes.

Services in the network have their starting and ending location modeled as nodes within G_t . For the service between two intermodal terminals $(i, j) \in \mathcal{A}_t^I$, there is a *maximum capacity* $Q_{i,j,t}$ measured in number of freights. For all services involving an origin or a destination, we assume that there is an unlimited number of trucks. All services $(i, j) \in \mathcal{A}_t$ have a *service duration* of $L_{i,j,t}^A$ days, which lasts at least one day. All transfer and handling operations at each location $i \in \mathcal{N}_t$ have a duration of $L_{i,t}^N$ days. To measure the total time required for the service between two locations, we define the parameter $M_{i,j,t} = L_{i,t}^N + L_{i,j,t}^A + L_{j,t}^N$. We assume that traveling directly to a destination by truck is always faster than going through an intermodal terminal, i.e., $L_{i,d,t}^A < \min_{j \in \mathcal{N}_t^I} \{M_{i,j,t} + L_{j,d,t}^A\}$, $\forall (i, d) \in \mathcal{A}_t^D$. This assumption works in a similar way as the triangle inequality in routing problems. All relevant costs from a service $(i, j) \in \mathcal{A}_t$ are captured in the *cost function* $C_{i,j,t}$. This means that, although pre- and end-haulage decisions, as well as freight handling decisions, are outside the scope of the planner, their costs can be captured with the function $C_{i,j,t}$.

Each day t , freights with different attributes become known to the planner. These freights are characterized by an origin $i \in \mathcal{N}_t^O$, a destination $d \in \mathcal{N}_t^D$, a release day $r \in \mathcal{R}_t = \{0, 1, 2, \dots, R_t^{\max}\}$, and a time-window length $k \in \mathcal{K}_t = \{0, 1, 2, \dots, K_t^{\max}\}$, where R_t^{\max} and K_t^{\max} are the maximum release day and time-window length, respectively, that a freight can have. Note that the absolute due-day is k days after r . Even though new freights and their characteristics are only known until they arrive, there is probabilistic knowledge about their arrival. In between two consecutive days $t - 1$ and t , a total of $f \in \mathbb{N}$ freights arrive into the system with probability $p_{f,t}^F$. A freight that arrives has origin $i \in \mathcal{N}_t^O$ with

probability $p_{i,t}^O$, destination $d \in \mathcal{N}_t^D$ with probability $p_{d,t}^D$, release-day $r \in \mathcal{R}_t$ with probability $p_{r,t}^R$, and time-window length $k \in \mathcal{K}_t$ with probability $p_{k,t}^K$.

3.2 MDP Model

In this section, we transform the problem horizon, constraints, and objective into the building blocks of an MDP: stages, state, decision, transition, and optimality equations. The stages of the MDP are defined by $t \in \mathcal{T}$. The state S_t consists of all freights in the network and their characteristics. To model these freights, we introduce the variable $F_{i,d,r,k,t} \in \mathbb{Z}^+$ that represents the number of freights at location $i \in \mathcal{N}_t^O \cup \mathcal{N}_t^I$, that have destination $d \in \mathcal{N}_t^D$, release day $r \in \mathcal{R}'_t$, and time-window length $k \in \mathcal{K}_t$; and define the state S_t as seen in (1). The state space is denoted as \mathcal{S} , i.e., $S_t \in \mathcal{S}$.

$$S_t = [F_{i,d,r,k,t}]_{\forall i \in \mathcal{N}_t^O \cup \mathcal{N}_t^I, d \in \mathcal{N}_t^D, r \in \mathcal{R}'_t, k \in \mathcal{K}_t} \tag{1}$$

Note that we use a new set \mathcal{R}'_t for the release days. The release day definition at origin nodes remains the same. The release day at an intermodal terminal, however, is now used to represent the days “left” for a freight to arrive at that node. For example, if a released freight is sent to an intermodal terminal j on a barge whose total service duration is four days, this freight will appear the day after it was sent, as a freight with $r = 3$ at location j . This new set, which is defined as $\mathcal{R}'_t = \left\{0, 1, 2, \dots, \max \left\{R_t^{\max}, \max_{(i,j) \in \mathcal{A}_t^I} M_{i,j,t}\right\}\right\}$, allows us to model multi-day durations of services without the need of remembering decisions from more than one day ago, i.e., to be more computationally efficient. Note that, in case no total service duration is larger than R_t^{\max} , then $\mathcal{R}_t = \mathcal{R}'_t$. Time-window lengths k still model the number of days after the release-day r , within which the freight has to be at its final destination. We will elaborate more on the evolution of the network over time later on in this section.

At each stage, the planner must decide how many released freights to transport and to postpone, for all locations. Remind that, in a synchronodal network, only the first part of the plan to transport a freight to its destination is implemented at each decision moment. Consequently, at every stage, the decision to transport a freight can be either to send it directly to its final destination, or to send it to an intermodal terminal. To model this decision, we introduce the variable $x_{i,j,d,k,t} \in \mathbb{Z}^+$, which represents the number of freights having destination $d \in \mathcal{N}_t^D$ and time-window length $k \in \mathcal{K}_t$ that are transported from location i to location j using service $(i, j) \in \mathcal{A}_t$. Thus, the decision vector x_t consists of all transported freights in the network, as seen in (2a).

$$x_t = [x_{i,j,d,k,t}]_{\forall (i,j) \in \mathcal{A}_t, d \in \mathcal{N}_t^D, k \in \mathcal{K}_t} \tag{2a}$$

s.t.

$$\sum_{j \in \mathcal{N}_t^I \cup \{d\}} x_{i,j,d,k,t} \leq F_{i,d,0,k,t}, \quad \forall i \in \mathcal{N}_t^O \cup \mathcal{N}_t^I, d \in \mathcal{N}_t^D, k \in \mathcal{K}_t \tag{2b}$$

$$x_{i,d,d,L_{i,d,t}^A,t} \geq F_{i,d,0,L_{i,d,t}^A,t}, \quad \forall (i,d) \in \mathcal{A}_t^D, k \in \mathcal{K}_t \quad (2c)$$

$$x_{i,j,d,k,t} = 0, \quad \forall (i,j) \in \mathcal{A}_t, d \in \mathcal{N}_t^D, k \in \mathcal{K}_t | k < M_{i,j,t} + M_{j,d,t} \quad (2d)$$

$$\sum_{d \in \mathcal{N}_t^D} \sum_{k \in \mathcal{K}_t} x_{i,j,d,k,t} \leq Q_{i,j,t}, \quad \forall (i,j) \in \mathcal{A}_t^I \quad (2e)$$

The decision x_t depends on the state S_t and the feasible decision space \mathcal{X}_t , which has four constraints. First, the number of freights transported from one location to all other locations cannot exceed the number of released freights available at the start location, as seen in (2b). Second, released freights whose time-window length is as long as the duration of direct transport (i.e., trucking) must be transported using this service, as seen in (2c). Third, freights whose time-window length is smaller than the duration of the shortest path between an intermodal terminal and their destination cannot be transported via that terminal, as seen in (2d). Fourth, transport between two intermodal terminals cannot exceed the capacity of the long-haul vehicle, as seen in (2e).

After making a decision x_{t-1} , but before entering the state S_t , new freights become known to the planner. We represent new freights with origin $i \in \mathcal{N}_t^O$, destination $d \in \mathcal{N}_t^D$, release day $r \in \mathcal{R}_t$, and time-window length $k \in \mathcal{K}_t$, by $\tilde{F}_{i,d,r,k,t}$. We denote the vector of all new freights that arrive between stages $t-1$ and t by W_t , as seen in (3). This vector represents the exogenous information (i.e., new random freights) that became known between stages $t-1$ and t .

$$W_t = \left[\tilde{F}_{i,d,r,k,t} \right]_{\forall i \in \mathcal{N}_t^O, d \in \mathcal{N}_t^D, r \in \mathcal{R}_t, k \in \mathcal{K}_t} \quad (3)$$

The evolution of the network over time is influenced by decisions, exogenous information, and various time relations. We represent this evolution by using a transition function S^M , as seen in (4a). The general idea of S^M is to define the freights at S_t using only the previous-stage decision x_{t-1} and the exogenous information W_t . Although decisions can span more than one day (i.e., when the duration of a service is longer than a day), we use freight release days (i.e., new set \mathcal{R}'_t) and time-window lengths to avoid remembering a decision for more than one stage. When freights are not transported, they remain at the same location and their release days and time-window lengths decrease. However, when freights are transported from a given location i to an intermodal terminal j , they are modeled as freights whose release day increases and their time-window length decreases in line with the total duration of transport $M_{i,j,t}$. To model all these relations, S^M classifies freight variables $F_{t,i,d,r,k}$ into seven categories, as shown in (4b)–(4h). To exemplify in detail the workings of these categories, consider (4c). These constraints apply to released freights at an intermodal terminal i with destination d and time-window length k . These freights are the result of three types of freights: (i) released freights in the same terminal, from the previous stage, that had the same destination, that had one additional day in the time-window, and that were not transported to any other node (i.e., $F_{t-1,i,d,0,k+1} - \sum_{j \in \mathcal{A}_t} x_{t-1,i,j,d,k+1}$); (ii) freights in the same node, from the previous stage, that had the same destination, that had a release-day of one, and that had the

same time-window length (i.e., $F_{t-1,i,d,1,k}$); and (iii) freights that arrived from other locations to i , that have the same destination, whose total duration of transportation was one period, and whose time-window length was $k + M_{j,i,t}$ at the moment of the decision x_{t-1} (i.e., $\sum_{j \in \mathcal{A}_t | M_{j,i,t}=1} x_{t-1,j,i,d,k+M_{j,i,t}}$). All other constraints work in a similar fashion.

$$S_t = S^M(S_{t-1}, x_{t-1}, W_t) \quad \text{s.t.} \quad (4a)$$

$$F_{t,i,d,0,k} = F_{t-1,i,d,0,k+1} - \sum_{j \in \mathcal{A}_t} x_{t-1,i,j,d,k+1} + F_{t-1,i,d,1,k} + \tilde{F}_{t,i,d,0,k},$$

$$\forall i \in \mathcal{N}_t^O, d \in \mathcal{N}_t^D, k+1 \in \mathcal{K}_t \quad (4b)$$

$$F_{t,i,d,0,k} = F_{t-1,i,d,0,k+1} - \sum_{j \in \mathcal{A}_t} x_{t-1,i,j,d,k+1} + F_{t-1,i,d,1,k}$$

$$+ \sum_{j \in \mathcal{A}_t | M_{j,i,t}=1} x_{t-1,j,i,d,k+M_{j,i,t}},$$

$$\forall i \in \mathcal{N}_t^I, d \in \mathcal{N}_t^D, k+1 \in \mathcal{K}_t \quad (4c)$$

$$F_{t,i,d,0,K_t^{\max}} = F_{t-1,i,d,1,K_t^{\max}} + \tilde{F}_{t,i,d,0,K_t^{\max}},$$

$$\forall i \in \mathcal{N}_t^O, d \in \mathcal{N}_t^D \quad (4d)$$

$$F_{t,i,d,r,k} = F_{t-1,i,d,r+1,k} + \tilde{F}_{t,i,d,r,k},$$

$$\forall i \in \mathcal{N}_t^O, d \in \mathcal{N}_t^D, r+1 \in \mathcal{R}_t | r \geq 1, k \in \mathcal{K}_t \quad (4e)$$

$$F_{t,i,d,r,k} = F_{t-1,i,d,r+1,k} + \sum_{j \in \mathcal{A}_t | M_{j,i,t}=r+1} x_{t-1,j,i,d,k+M_{j,i,t}},$$

$$\forall i \in \mathcal{N}_t^I, d \in \mathcal{N}_t^D, r+1 \in \mathcal{R}'_t | r \geq 1, k \in \mathcal{K}_t \quad (4f)$$

$$F_{t,i,d,|\mathcal{R}'_t|,k} = \sum_{j \in \mathcal{A}_t | M_{j,i,t}=|\mathcal{R}'_t|+1} x_{t-1,j,i,d,k+M_{j,i,t}}$$

$$\forall i \in \mathcal{N}_t^I, d \in \mathcal{N}_t^D, k \in \mathcal{K}_t, \quad (4g)$$

$$F_{t,i,d,R_t^{\max},k} = \tilde{F}_{t,i,d,R_t^{\max},k},$$

$$\forall i \in \mathcal{N}_t^O, d \in \mathcal{N}_t^D, k \in \mathcal{K}_t \quad (4h)$$

The goal is to minimize the total costs over a multi-period horizon, considering all possible states that can occur each day, and considering the stochastic arrival of freight. To do so, we define a policy π as a function that maps each possible state $S_t \in \mathcal{S}$ to a decision $x_t^\pi \in \mathcal{X}_t$. Consequently, the objective is to determine the policy π from the set of all policies Π that minimizes the expected costs over the planning horizon, given an initial state S_0 , as seen in (5):

$$\min_{\pi \in \Pi} \mathbb{E} \left[\sum_{t \in \mathcal{T}} C_t(x_t^\pi) = \sum_{t \in \mathcal{T}} \sum_{(i,j) \in \mathcal{A}_t} \left(C_{i,j,t} \cdot \sum_{d \in \mathcal{N}_t^D} \sum_{k \in \mathcal{K}_t} x_{i,j,d,k,t}^\pi \right) \middle| S_0 \right] \quad (5)$$

To solve this stochastic and sequential optimization problem, we transform (5) into the Bellman’s equations (6). In these equations, the expected next-stage cost is computed using the value of the next-stage state S_{t+1} (obtained using S^M), the decision x_t^π , a realization of the exogenous information $\omega \in \Omega_{t+1}$, and the associated probability $p_\omega^{\Omega_{t+1}}$. The solution to all recursive equations of (6), e.g., through backward induction, provide the optimal policy for the MDP.

$$V_t(S_t) = \min_{x_t^\pi \in \mathcal{X}_t} \left(C_t(x_t^\pi) + \sum_{\omega \in \Omega_{t+1}} p_\omega^{\Omega_{t+1}} \cdot V_{t+1}(S^M(S_t, x_t^\pi, \omega)) \right), \forall S_t \in \mathcal{S} \quad (6)$$

However, solving the Bellman equations (6) for large problems is computationally challenging. The state space \mathcal{S} , decision space \mathcal{X}_t , and the realizations of the exogenous information in Ω_t grow larger with an increasing size of the problem instance. Due to these three “curses of dimensionality” [12], our MDP model is solvable only for tiny problem instances. Notwithstanding, the MDP model serves as a base for our ADP approach.

4 Solution Approach

Our solution approach is based on ADP, which is a heuristic solution method for MDP models that uses various constructs and algorithmic strategies. Its main idea is to replace the expected next-stage costs in (6) by a Value Function Approximation (VFA), and to update this function via a simulation of the exogenous information. This update procedure is done iteratively, as shown in Fig. 2, with the end result being the approximated values of the solution to the Bellman’s equations, and thus a policy π . Certainly, the choice of (i) VFA, (ii) the update procedure, and (iii) the number of iterations, has an influence on the performance, i.e., solution quality and computational time. In the following paragraphs we describe the choices we make and indicate their expected performance.

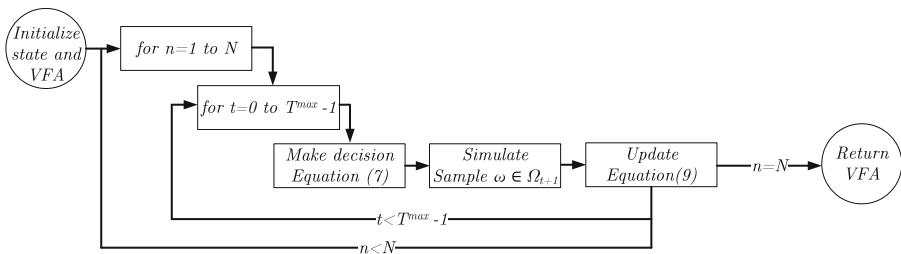


Fig. 2. Overview of the ADP algorithm

In our ADP approach, the expectation of future costs in (6) is replaced by an value function approximation $\bar{V}_t^n(S_t^{x,n})$, where $S_t^{x,n}$ is the so-called post-decision

state, i.e., the state after a decision has been made but before the new exogenous information becomes known. As seen in (7), this construct avoids specifying all realizations of the exogenous information Ω_t .

$$V_t^n(S_t^n) = \min_{x_t^\pi \in \mathcal{X}_t} \left(C_t(x_t^\pi) + \bar{V}_t^n(S_t^{x,n}) \right) \tag{7}$$

To avoid the large state space, the optimality equation in (7) is solved for one state at each stage, starting from the initial state S_0 . The transition from one state to the next uses a sample from Ω_{t+1} , obtained through a Monte Carlo simulation, and the transition function S^M defined in (4a). This process is performed for the entire planning horizon, and repeated for N iterations, hence the superscript n in the approximate value function and post-decision state.

The general outline of an ADP algorithm can be found in Fig. 4.7, page 141, of the book of [12]. We now focus on two designs (i.e., variations) we propose for that algorithm. Our first design uses a commonly proposed ADP setup. We use *basis functions* for $\bar{V}_t^n(S_t^{x,n})$ and the *non-stationary least squares* method for updating this function. A basis function $\phi_a(S_t^{x,n})$ is a quantitative characteristic of a given feature a of a post-decision state $S_t^{x,n}$ that describes, to some extent, the value of that post-decision state. Examples of features in our problem are the number of freights for a given destination and the number of freights at a given intermodal terminal. Given a set of features \mathcal{A} , the approximated next-stage costs in (7) are the result of the product between the basis function $\phi_a(S_t^{x,n})$ and the weight $\theta_{a,t}^n$ for each feature $a \in \mathcal{A}$, as seen in (8).

$$\bar{V}_t^{x,n}(S_t^{x,n}) = \sum_{a \in \mathcal{A}} \theta_{a,t}^n \phi_a(S_t^{x,n}) \tag{8}$$

The weight $\theta_{a,t}^n$ depends on the iteration n because it is updated after each iteration, using observed costs, to improve future cost estimates. We use a Non-stationary Least Squares (NLS) method for updating these weights since it gives more emphasis to the recent observation than to the previous one. This emphasis is necessary at early iterations, where initial conditions might bias the approximation and the result of the ADP approach. The weights $\theta_{a,t}^n$, for all $a \in \mathcal{A}$, are updated each iteration n using the observed error (i.e., difference between the next-stage estimate from the previous iteration $\bar{V}_{t-1}^{n-1}(S_{t-1}^{x,n})$ and the current estimate \hat{v}_t^n), the value of all basis functions $\phi_a(S_t^{x,n})$, the optimization matrix H^n , and the previous weights $\theta_{a,t}^{n-1}$, as seen in (9). For a comprehensive explanation on the NLS method, we refer to [12].

$$\theta_{a,t}^n = \theta_{a,t}^{n-1} - H_n \phi_a(S_t^{x,n}) \left(\bar{V}_{t-1}^{n-1}(S_{t-1}^{x,n}) - \hat{v}_t^n \right) \tag{9}$$

The first design considers downstream costs only through a one-step estimate. Since estimates can be off, especially in early iterations, it might be beneficial to look ahead more than one step. To do this, our second design builds on the first one and uses two additional constructs. First, we add a valid inequality to the decision space \mathcal{X}_t as follows. If a direct service for a freight between

its origin and its destination is cheaper than going from its origin to a given intermodal terminal and subsequently to its destination, we prevent this freight from going to that intermodal terminal when its time-window length allows only a direct service after the intermodal terminal. Second, we add another estimate to $\bar{V}_t^n(S_t^{x,n})$, as seen in (10). In this new approximate value function, $\bar{C}_t^n(S_t^{x,n})$ is an estimate of the downstream cost obtained simulating a fixed rule for the remaining of the horizon, under different demand realizations, and α is a weight to balance the use of basis functions and simulations for $\bar{V}_t^n(S_t^{x,n})$.

$$\bar{V}_t^{x,n}(S_t^{x,n}) = \alpha \sum_{a \in \mathcal{A}} \theta_{a,t}^n \phi_a(S_t^{x,n}) + (1 - \alpha) \bar{C}_t^n(S_t^{x,n}) \quad (10)$$

At last, the output of our two ADP designs are the weights $\theta_{a,t}^N$. The resulting policy π maps state $S_t \in \mathcal{S}$ to decision x_t^π as seen in (11).

$$x_t^\pi = \arg \min \left(C_t(x_t^\pi) + \sum_{a \in \mathcal{A}(S_t^x)} \theta_{a,t}^N \phi_a(S_t^x) \right) \quad (11)$$

5 Numerical Experiments

In this section, we explore the value of our ADP designs through a series of numerical experiments. Using three small instances, we compare the costs achieved by our ADP approach against a benchmark policy and an advance sampling procedure. The benchmark policy mimics a planning approach commonly used in practice. The sampling procedure extends the benchmark policy with a methodology commonly considered in the literature. The section is divided as follows. First, we introduce our experimental setup. Second, we show, analyze, and discuss the results of our experiments.

5.1 Experimental Setup

For the three instances, we use a network containing a single origin, three intermodal terminals, and three destinations over a planning horizon of 15 days. Each day, there are three services between the intermodal terminals, with capacities and durations as shown in Fig. 3. The fixed costs of these services are of $C_{1,2}^F = C_{2,3}^F = 100$ and $C_{1,3}^F = 150$. The variable costs range between 36 and 44, and are equal to the Euclidean distance between the terminals in a plane of 100×50 distance units, as shown to scale in Fig. 3. In addition, every day there is a direct service between the origin and the terminals, between the origin and the destinations, and between the terminals and the destinations; and they all have duration of one day. There are no fixed costs for the direct services, and the variable cost range varies between 241 and 927, and are equal to ten times the Euclidean distance between the two locations they connect. The number of freights that arrives each day varies between $f = \{0, 1, \dots, 4\}$, with probability p_f^F as shown in Fig. 3. In the three instances, each freight has destination

$d \in \{4, 5, 6\}$ with probability p_d^D as shown in Fig. 3, and is always released (i.e., $p_0^R = 1$). Each freight has a time-window length $k = \{1, 2, \dots, 5\}$ with probability p_k^K according to the instances considered. In instances where freights have short time-windows, there are not many feasible options for transportation and almost none for postponement. In instances with large time-windows, the opposite occurs. To test the value of look-ahead decisions, we create instances with different time-window length distributions, as shown in Fig. 3.

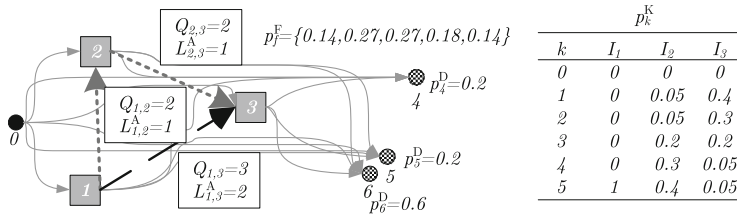


Fig. 3. Network characteristics for the test instances

Using these instances, we test four planning methods: a benchmark heuristic, our two ADP designs (named ADP 1 and ADP 2), and an advance sampling procedure. The set of features \mathcal{A} consists of all state variables and a constant of 1. The weight α for ADP 2 is defined as $\alpha = \max\{25/(25 + n - 1), 0.05\}$ and the sampling method is the same as the advance sampling procedure introduced in the next paragraph. The number of iterations is set to 100 and the NLS parameters used are those recommended by [11]. Although these settings achieved a fast convergence of the ADP algorithm in our tests, the resulting approximate value functions (i.e., policy) is heuristic and not necessarily optimal.

The benchmark heuristic strikes for a balance between using the intermodal services efficiently (consolidate as many freights as possible) and the postponement of freight. It consists of four steps: (i) define the shortest and second shortest path for each freight to its final destination, without considering fixed costs for services between terminals, (ii) calculate the savings between the shortest and second shortest path and define these as savings of the first intermodal service used in the shortest path, (iii) sort all freights in non-decreasing time-window length, i.e., closest due-day first, and (iv) for each freight in the sorted list, check whether the savings of the first intermodal service of its shortest path are larger than the fixed cost for this service; if so, use this service for the freight, if not, postpone the transport of the freight. Naturally, all capacities, durations, and time-windows must be checked while doing these steps.

The sampling procedure consists of three steps: (i) enumerate all feasible decisions, (ii) for each feasible decision, estimate future costs by sampling, in a Monte Carlo fashion and using common random numbers across the decisions, realizations of the exogenous information for the remainder of the planning horizon, and simulating the use of the benchmark heuristic for making decisions with these samples, and (iii) choose the decision with the lowest sum of direct and

estimated future costs. Although heavily computationally intensive (i.e., not applicable to larger instances), this procedure exploits the benefits of looking-ahead in decision making.

The tests are done using ten test states in each instance. To define these states for each instance, we do a simulation of the benchmark heuristic, beginning with an empty state, for a horizon of 15 days. We save the state at the end of the horizon. We replicate this procedure 10,000 times, and choose the ten states that were observed the most. For each of the test states, we simulate each planning method 100 times, using common random numbers across the methods. Note that these 100 simulation replications are different from the 100 iterations of the ADP algorithm. Thus, we test the ADP approach in two phases: (i) learning phase through 100 iterations and (ii) simulation phase of using the resulting policy in (11) for 100 replications.

5.2 Experimental Results and Discussion

First, we analyze Instance I_1 . This is the most flexible test instance since all freights have a time-window length of 5 days when they arrive. The results are shown in Table 1. We show the costs for the benchmark heuristic, and the relative savings, as a percentage, of the other planning methods when compared to the benchmark. In addition, we show the number of freights of each test state and the computational time. The computational time (in seconds) is given as the total simulation time for the 100 replications of the 15-day horizon.

Table 1. Results for Instance I_1

State	Total Freights	Benchmark		ADP 1		ADP 2		Sampling	
		Solution	Time (s)	Solution	Time (s)	Solution	Time (s)	Solution	Time (s)
1	4	12221	0.92	-13.6%	29.94	-33.9%	101.31	-43.3%	688.29
2	7	14684	0.94	-12.8%	52.06	-32.7%	96.67	-39.9%	1687.18
3	5	13042	0.92	-13.1%	31.68	-27.5%	81.46	-41.5%	827.15
4	6	13863	0.94	-12.3%	32.99	-25.9%	81.42	-39.0%	832.67
5	6	13863	0.91	-12.0%	108.62	-30.0%	111.21	-42.3%	1356.80
6	6	13863	0.94	-10.4%	102.12	-31.3%	67.58	-42.9%	1317.73
7	5	13042	0.94	-12.6%	40.26	-23.4%	81.99	-41.5%	893.59
8	4	12221	0.92	-14.7%	37.44	-25.0%	78.41	-38.9%	547.66
9	2	10579	0.94	-14.9%	31.72	-29.9%	45.13	-42.4%	611.18
10	5	13042	1.01	-11.2%	30.81	-32.9%	42.92	-40.6%	727.28

On average, ADP 1 achieves savings of 12.8%, ADP 2 of 29.2%, and the sampling procedure of 41.2% when compared to the benchmark heuristic. All three methods that explicitly look-ahead in their decisions perform better than the benchmark that does so only implicitly. The sampling method performs the best, at a higher computational expense (more than 10 times the computational time of ADP, and 1000 times the one of the benchmark, on average). For large

instances, or even small ones where time is discretized into smaller intervals, this method would not be applicable. ADP 2 performs second best, at a higher computational expense during the learning phase than ADP 1 (965s instead of 116, on average). However, during the actual decision making for the entire planning horizon, both ADP designs have similar computational time (50 and 79 s, on average, respectively, for I_1). ADP 1 lowest savings indicate that a one-step look-ahead is not sufficient for a good solution. Furthermore, the difference between the two ADP designs suggests that further research that explicitly considers a few stages in advance, such as rolling-horizon procedures within the ADP framework, can improve performance significantly.

The average results across the test states of I_2 and I_3 are shown in Table 2. Note that each instance has its own set of test states, which differs from the other instances. Furthermore, note that I_2 and I_3 have significantly less flexibility than I_1 due to their time-window length, only 40 % and 0.05 % of arriving freights can use any intermodal connection, respectively.

Table 2. Average results for Instance I_2 and I_3

Instance	Benchmark		ADP 1		ADP 2		Sampling	
	Solution	Time (s)	Solution	Time (s)	Solution	Time (s)	Solution	Time (s)
I_2	11078	0.88	-5.2 %	10.52	-9.8 %	13.89	-31.2 %	217.19
I_3	12874	1.01	2.9 %	3.19	0.4 %	2.31	-3.3 %	36.95

The larger savings from all look-ahead methods in I_1 and I_2 , compared to I_3 , indicate that the more flexibility there is, the better it is to look-ahead when making decisions. In I_2 , similar results to I_1 are achieved, but with significantly less cost savings. In I_3 , the benchmark heuristic performs better than the ADP approach, and the sampling achieves small savings. In most states of I_3 , the only feasible option (time-wise) for freights is to use a direct service via truck. In such a setting, decision making methods that focus on current costs, such as the benchmark heuristic, perform well since there are hardly consolidation opportunities to anticipate for. However, a robust ADP design should be able to learn such a policy, as the sampling method seems to do. In a sensitivity analysis (results not shown), we observed that the number of iterations and the NLS parameters had a small impact on the solution quality, compared to the impact of different approximate value functions (e.g., more basis functions, the sampling method, etc.). In a similar way, the different approximate value functions had a significant difference in computational time during the learning phase, but not during the decision-making phase. Further research on adaptations of the approximating function of future costs within the ADP algorithm, such as aggregates, hierarchical functions and state representatives, is necessary.

6 Conclusions

We developed an MDP model and an ADP algorithm for selecting services and transfers for freights in a sychromodal network. With the MDP model, the optimal balance between transporting and postponing freights, in different locations of the network, over time, and under uncertain demand, can be achieved. With the ADP algorithm, the computational burden of the MDP model is reduced while preserving all of its modeling functionalities.

Through numerical experiments, we explored the value of using look-ahead decisions in our planning problem and reflected on the value and the limitations of our ADP designs. We observed that the more time-window flexibility and number of freights there are, the better the look-ahead methods perform. We also observed that the two methods that look-ahead more than one stage performed better than the standard one-step look-ahead ADP approach. Further research about ADP designs that explicitly consider a few stages in advance (e.g., rolling horizon, sampling, approximate policy iteration) and other, possibly non-linear, value function approximations, are relevant for sychromodal planning.

References

1. Bai, R., Wallace, S.W., Li, J., Chong, A.Y.L.: Stochastic service network design with rerouting. *Transp. Res. Part B* **60**, 50–65 (2014)
2. Caris, A., Macharis, C., Janssens, G.K.: Decision support in intermodal transport: a new research agenda. *Comput. Ind.* **64**(2), 105–112 (2013). Decision Support for Intermodal Transport
3. Crainic, T.G., Hewitt, M., Rei, W.: Scenario grouping in a progressive hedging-based meta-heuristic for stochastic network design. *Comput. Oper. Res.* **43**, 90–99 (2014)
4. Dall’Orto, L.C., Crainic, T.G., Leal, J.E., Powell, W.B.: The single-node dynamic service scheduling and dispatching problem. *Eur. J. Oper. Res.* **170**(1), 1–23 (2006)
5. Ghane-Ezabadi, M., Vergara, H.A.: Decomposition approach for integrated intermodal logistics network design. *Transp. Res. Part E* **89**, 53–69 (2016)
6. Li, L., Negenborn, R.R., Schutter, B.D.: Intermodal freight transport planning a receding horizon control approach. *Transp. Res. Part C* **60**, 77–95 (2015)
7. Lium, A.G., Crainic, T.G., Wallace, S.W.: A study of demand stochasticity in service network design. *Transp. Sci.* **43**(2), 144–157 (2009)
8. Lo, H.K., An, K., Lin, W.H.: Ferry service network design under demand uncertainty. *Transp. Res. Part E* **59**, 48–70 (2013)
9. Mes, M.R.K., Iacob, M.E.: Sychromodal transport planning at a logistics service provider. In: Zijm, H., Klumpp, M., Clausen, U., ten Hompel, M. (eds.) *Logistics and Supply Chain Innovation: Bridging the Gap Between Theory and Practice*, pp. 23–36. Springer, Heidelberg (2016)
10. Nabais, J., Negenborn, R., Bentez, R.C., Botto, M.A.: Achieving transport modal split targets at intermodal freight hubs using a model predictive approach. *Transp. Res. Part C* **60**, 278–297 (2015)
11. Rivera, A.P., Mes, M.: Dynamic multi-period freight consolidation. In: Corman, F., Voß, S., Negenborn, R.R. (eds.) *ICCL 2015. LNCS*, vol. 9335, pp. 370–385. Springer, Heidelberg (2015). doi:[10.1007/978-3-319-24264-4_26](https://doi.org/10.1007/978-3-319-24264-4_26)

12. Powell, W.B.: *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, 2nd edn. Wiley, Hoboken (2011)
13. Riessen, B., Negenborn, R.R., Dekker, R.: Synchronodal container transportation: an overview of current topics and research opportunities. In: Corman, F., Voß, S., Negenborn, R.R. (eds.) *ICCL 2015. LNCS*, vol. 9335, pp. 386–397. Springer, Heidelberg (2015)
14. SteadieSeifi, M., Dellaert, N., Nuijten, W., Woensel, T.V., Raoufi, R.: Multimodal freight transportation planning: a literature review. *Eur. J. Oper. Res.* **233**(1), 1–15 (2014)
15. Wieberneit, N.: Service network design for freight transportation: a review. *OR Spectrum* **30**(1), 77–112 (2008)
16. Zhang, M., Pel, A.: Synchronodal hinterland freight transport: model study for the port of Rotterdam. *J. Transp. Geogr.* **52**, 1–10 (2016)