



TAPAS: A multi-agent-based model for simulation of transport chains

Johan Holmgren^{a,*}, Paul Davidsson^{a,b}, Jan A. Persson^{a,b}, Linda Ramstedt^c

^a School of Computing, Blekinge Institute of Technology, Karlskrona, Sweden

^b School of Technology, Malmö University, Malmö, Sweden

^c Vectura, Svetsarvägen 24, Box 46, SE-171 11 Solna, Sweden

ARTICLE INFO

Article history:

Received 26 October 2010

Received in revised form 12 September 2011

Accepted 29 December 2011

Available online 31 January 2012

Keywords:

Multi-agent-based simulation

Multi-agent systems

Transport chain simulation

Freight transportation

Decision-making

ABSTRACT

We present the Transportation And Production Agent-based Simulator (TAPAS), which is an agent-based model for simulation of transport chains that can be used, e.g., for analysis of transport-related policy and infrastructure measures. TAPAS is more powerful than traditional approaches to freight transport analysis, as it explicitly models production and customer demand, and it captures the interaction between individual transport chain actors, their heterogeneity and decision making processes, as well as time aspects. Whereas traditional approaches rely on assumed statistical correlation, TAPAS relies on causality, i.e., the focus is on the decisions and negotiations that lead to activities. TAPAS is composed of two connected layers, one that simulates the physical activities, e.g., production and transportation, and one that simulates the decision making and interaction between actors. We illustrate TAPAS with a scenario in which the consequences of three transport policy and infrastructure measures are studied.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

Freight transportation causes different types of positive and negative effects on the society. Positive effects typically relate to economy and social welfare, e.g., due to the possibility to consume products that have been produced far away. Negative effects mainly relate to the environment, and typical examples are emissions, congestion and energy use. By applying different types of governmental control policies, infrastructure investments and strategic business strategies, it is sometimes possible to influence how transportation and other activities in transport chains are performed. Control policies include different types of taxes and fees, such as kilometer and fuel taxes, and regulations, such as weight restrictions on vehicles. Infrastructure investments can be made into roads, railway tracks, intermodal freight terminals, industry tracks, etc., and examples of strategic business measures are improvement of timetables and adjustment of vehicle fleets to better meet the demand for transport. Public authorities, in the role of policy makers, often have a wish to reach certain governmental goals, such as obtaining sustainable transport systems and meeting emission targets. A typical ambition of a public authority is to increase the internalization of external costs, e.g., by letting road users pay for the road wear they cause [7]. However, internalization of external costs may have effects that might be negative on other goals. For instance, it might lead to negative economic development in a region. For enterprises, the goal is typically to maximize profit, e.g., through optimization of their activities (either individually or in collaboration), by reducing lead-times, lowering transport costs, improving delivery accuracy, etc. To avoid undesired consequences and to confirm desired consequences it is important to be able to accurately predict what the consequences will be when applying different types of measures.

* Corresponding author. Address: School of Computing, Blekinge Institute of Technology, Box 214, SE-374 24 Karlskrona, Sweden.

E-mail addresses: johan.holmgren@bth.se (J. Holmgren), paul.davidsson@bth.se, paul.davidsson@mah.se (P. Davidsson), jan.persson@bth.se, jan.a.persson@mah.se (J.A. Persson), linda.ramstedt@vectura.se (L. Ramstedt).

Multi-Agent-Based Simulation (MABS), in which one or more of the simulated entities are modeled as agents, can be used for studying freight transport systems. An agent can be viewed as a system that is situated in some environment and capable of autonomous action in that environment in order to meet its design objectives [47]. We present the Transportation And Production Agent-based Simulator (TAPAS), which is a micro-level model for quantitative impact assessment of, e.g., different types of transport-related policy and infrastructure measures. In TAPAS, which is based on agent technology, the individual actors of a transport chain, e.g., producers, transport operators and customers, are explicitly modeled. TAPAS incorporates the complexity of transport choices, e.g., with respect to consignment size, and route and transport mode. In addition, TAPAS is able to take into account time aspects, such as, the effects of timetables, arrival times, and time-differentiated taxes and fees. This makes TAPAS more powerful than traditional approaches to freight transport simulation as it is able to capture the individual goals of transport chain actors and the interaction between them, as well as their heterogeneity and decision making. Whereas traditional approaches rely on assumed statistical correlation between different parameters, TAPAS relies on causality, i.e., the decisions and negotiations that lead to activities are modeled.

The main application area for TAPAS is to provide decision support to public policy makers by enabling analysis of, e.g., different types of transport-related policy and infrastructure measures. Other possible applications of TAPAS include assisting companies in making tactical and operational decisions, e.g., concerning choice of consignment size and frequency of deliveries, as well as strategic decisions, e.g., concerning choice of producer, adaptation of vehicle fleets, and location of storages and terminals. Moreover, we argue that TAPAS may complement existing approaches (e.g., on the macro-level) in different ways, e.g., by generating transport demand that may serve as input data to other models.

The research concerning TAPAS has been going on for several years, and the work has rendered a number of conference publications (e.g., [5,21,28,13]). This paper extends the previous work by providing a more detailed technical description and better motivation of the design of the TAPAS simulation model. Also, TAPAS has been extended with new functionality, e.g., load consolidation for demand driven transports has been added, and the model for setting transport prices has been improved.

In the next section we discuss some related work, followed in Section 3 by a detailed description and motivation of the TAPAS simulation model, including an account to the implementation of TAPAS. In Section 4 we illustrate some important functionalities provided by TAPAS by presenting a simulation study that has been conducted using TAPAS. Finally, in Section 5 and Section 6, we conclude the paper with a discussion and some suggestions for future work.

2. Related work

In this paper we present TAPAS, which is an agent-based simulation model that can be used, e.g., for impact assessment of transport-related policy and infrastructure measures and evaluation of different types of strategic business strategies. TAPAS is best described as a transport chain simulator, which has been extended with functionality for simulating production. In TAPAS, transportation is modeled in more detail than production, and it only considers a single-tier of the supply chain. Therefore, TAPAS mainly relates to the transport domain, and only partially to the more production-focused supply chain domain. Hence, the work behind TAPAS can be regarded as multi-disciplinary, and we will in this section cover several domains. We focus on research related to freight transport analysis and we also pay some attention to research related to supply chain modeling, focusing on agent-based supply chain models. It should be mentioned that the focus of this section is to give an overview of relevant work, not to conduct a complete literature review since reviews in relevant domains already exist. For instance, Terzi and Cavalieri [42] contributed with a review focusing on parallel and distributed supply chain simulation. In the transport modeling domain, several reviews exist (e.g., [12,15,40]), and Friedrich and Liedtke [18] provided a review of current transport modeling literature focusing on two agent-based models.

TAPAS simulates freight transportation, where the focus is on the freight flows and its allocation to vehicles, and models with this focus is often called commodity flow based models. Another related type of model is tour-based models, which instead focus on vehicle movements, i.e., the focus is on tours in which vehicles often start and end in the same location. An example of a tour-based model is the Calgary model presented by Hunt and Stefan [22]. Tour-based models are more often used when the geographical focus is regional or urban, while commodity flow based models often have a national or international focus.

Traditionally, freight transportation has been studied using macro-level (or equation-based) models, such as Samgods [38], ASTRA [32], and SISD [45]. Models of this type take a societal perspective and are based on aggregated coarse-grained data on national or regional levels, and different countries typically use their own models. Moreover, such models are commonly based on the aggregate and sequential four-step approach. Models based on this approach include one or several sub-models that can be described using four steps; production & attraction (trip generation), trip distribution, modal choice, and aggregation [15]. Models based on the four-step approach can look rather different, and different steps are typically represented by different types of sub-models, such as system dynamics models (e.g., ASTRA [32]) and economic models such as I/O models (e.g., Samgods [38]) and gravity models (e.g., NEAC [11]). An overview of models based on the four-step approach is provided in [15]. Further, sometimes more than one step is represented within the same sub-model, such as in Samgods [38], in which modal choice and assignment is integrated. An issue with traditional aggregate sequential models is that they do not take the logistical processes into account, e.g., choices of carrier type and ordering strategies, and thus fail to model the level where decisions regarding the actual transports take place. Starting about a decade ago, several aggregate sequential models, which also take logistical aspects into consideration, have been developed. Such models are for example SLAM [45],

SMILE [41], GoodTrip [6], and the model suggested by de Jong and Ben-Akiva [14], and they typically make use of disaggregation of aggregate data. However, since these models do not represent individual transport chain actors, the complex interactions (e.g., negotiation) and the decision making processes from which transport plans result are impossible to capture. Moreover, these models can only capture average times, but typically no aspects related to dynamic time (e.g., timetabled transportation), which is crucial when logistical decisions are coordinated. The decision making in supply chains is subject to both short-term and long-term planning implying that the time dimension of the decisions needs to be considered.

We argue that more precise predictions regarding the effects of transport-related policy and infrastructure measures, and strategic business measures, can be achieved by using micro-level, and in particular agent-based models. Such models enable to also capture the decision making of actors involved in logistical processes. This is important for the ability to capture the causality from which transport activities appear. Agent-based models are often also designed in a way that they explicitly model time aspects.

Since TAPAS is an agent-based model for transport chain simulation, we will in the rest of this section discuss some agent-based models for supply chain and freight transport analysis. As an example, an agent-based supply chain modeling framework with the purpose of assisting managers to analyze and understand the impact (in terms of costs, risks, etc.) of different decisions was presented in [39]. The framework made use of a library of reusable software components to promote reusability, and it was used for analyzing different inventory policies with the purpose of achieving improved inventory management. The similarity TAPAS has with supply chain models like the one in [39] is that it is built around similar structural components, however, the application area of TAPAS is similar to that of traditional freight transport models. In supply chain models, common structural components are activities/processes (such as manufacturing and assembling) and actors/roles (such as manufacturer and customer). Strader et al. [37] presented a multi-agent-based simulation tool, which was used to show the importance of information technology for supporting the order fulfillment process. This was achieved by studying a 5-tier supply chain network under different demand management and information sharing policies. In a simulation model by Gjerdrum et al. [19], agent technology and mathematical optimization were combined in order to study different inventory replenishment policies. Mathematical optimization was used to generate locally optimal manufacturing plans, and agents (e.g., customer, warehouse, factory and transport agents) were used to manage the supply chain. In comparison to our approach, the model by Gjerdrum et al. models manufacturing on a more detailed level, but transportation is not modeled on a detail that allows transport-related policies to be studied. Liedtke [26] presented an agent-based approach to commodity modeling (INTERLOG), which is relevant for our work since its aim is to study the same types of questions as can be studied with TAPAS. The major difference between the models concerns which actors dominate supply chains; in TAPAS the customer is dominating, whereas in INTERLOG the supply chain is dominated by the producers and the transport operators. Moreover, in contrast to TAPAS, INTERLOG assumes a pre-determined, however stochastically generated, transport demand. This makes it impossible to capture changes in transport demand, e.g., caused by changes in logistical structures. Moreover, the literature contains a number of newer agent-based simulation approaches [44,10,29], which are presented as re-usable non-implemented conceptual frameworks that can be used as guidance when constructing agent-based simulation models. While these models are more on a conceptual level, TAPAS is more concrete as it is an implemented model.

In summary, the literature contains a number of agent-based models for supply chain and freight transport analysis, however most of them focus on simulating already established chains, while TAPAS partially constructs the transport chain, i.e., product supplier, route and transport operators are dynamically chosen. This is an advantage, since it enabled us to base the TAPAS model on supply and demand for products instead of transport demand, which is the case in most existing freight transport models.

3. The simulation model

In this section we present the TAPAS simulation model, which has a two-tier architecture with a *physical simulator* and a *decision making simulator*, as illustrated in Fig. 1. In the physical simulator, all physical entities (e.g., vehicles, production facilities, transportation infrastructure) are modeled, and in the decision making simulator, a set of transport chain decision makers (or roles) are modeled as agents. The main motivation for choosing a two-tier architectural design is that entities in the physical simulator are considered passive, while entities in the decision making simulator act independently and potentially proactively. The two layers are connected in a way that the activities in the physical simulator are initiated by the decisions taken by the decision makers. The agents appear in a hierarchical structure (see Fig. 1) to be able to represent the hierarchical decision making structure that is common in real-world transport chains. In hierarchical decision making structures, information is not automatically made available for all agents, which may result in a certain degree of local optimization. This appears to be rather typical in transport chains today, since often only a limited amount of information is shared [49].

TAPAS has mainly been validated by interviews with experts in policy issues and transport modeling, and with practitioners in transportation and logistics. Several simulation studies have been performed with TAPAS (e.g., [13]), in which the sensitivity of different input parameters has been analyzed, and the results have in some studies been compared to similar studies [27].

3.1. The physical simulator

TAPAS uses a transport network, which can be described as a directed graph (N, L) with a set of nodes N and a set of directed links L . A node can be a customer node, a producer (factory) node, or a connection point (terminal) in the network. For

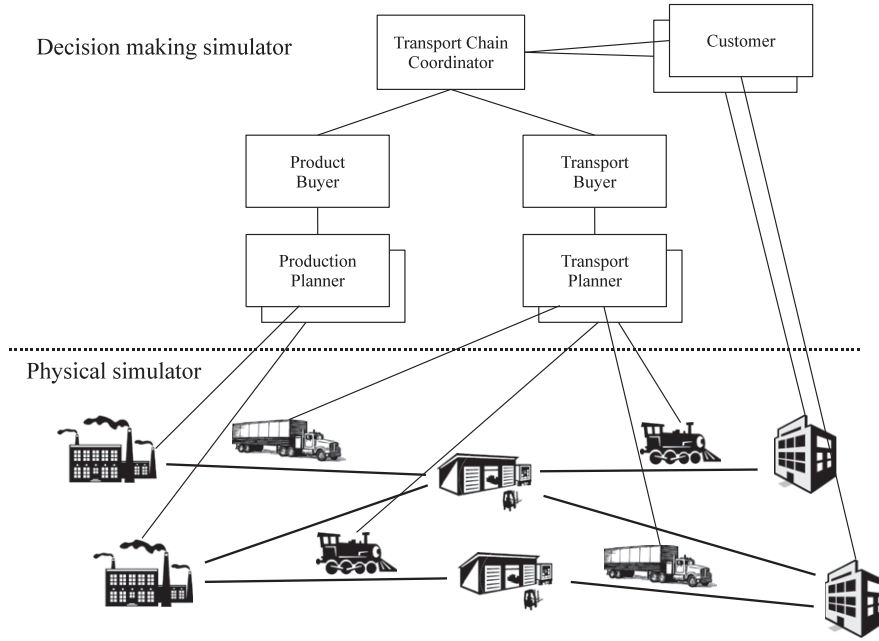


Fig. 1. Overview of the TAPAS simulation model with a physical simulator that is connected to a decision making simulator.

future reference, we let $N^C \subseteq N$ refer to the customer nodes and $N^F \subseteq N$ to the producer nodes. A link represents a directed connection between two network nodes and it corresponds to exactly one transport mode (typically road, rail or sea), and two nodes may be connected by multiple links representing different modes.

3.1.1. Production

Commonly used production strategies are pull and push, where production is demand driven in the former and forecast driven in the latter. Pull strategies are often used to decrease inventory levels and to better capture demand variations, while push strategies often use resources better due to the potential for improved planning, e.g., by producing larger batches [1]. In TAPAS we have chosen a pull strategy, in which products are produced in batches.

TAPAS models a set of product types P , and $P_{n^f} \subseteq P$ denotes the set of product types that can be produced in node n^f . For each product type $p \in P_{n^f}$, we let $c_{n^f p}^{mtrl}$ denote the raw material cost per unit, $s_{n^f p}^{batch}$ the maximum batch size, $c_{n^f p}^{prod\ time}$ the production cost per time unit, and $t_{n^f p}^{batch\ proc}$ the batch processing (production) time. A setup time $t_{n^f p}^{batch\ setup}$ (e.g., representing changeover) is considered when two batches of different product types are produced in sequence. The expected time

$$t_{n^f p}^{prod}(q) = \left\lceil q / s_{n^f p}^{batch} \right\rceil \cdot (t_{n^f p}^{batch\ setup} + t_{n^f p}^{batch\ proc}) \quad (1)$$

for producing q units of product type p in node n^f is calculated as the number of batches times the batch production time, where the setup time is considered only if the previously produced batch had a different product type. The expected cost

$$c_{n^f p}^{prod\ cost}(q) = t_{n^f p}^{prod}(q) \cdot c_{n^f p}^{prod\ time} + c_{n^f p}^{mtrl} \cdot q, \quad (2)$$

for producing q units of product type p is calculated as the production cost plus the cost for raw material.

Hence, TAPAS makes use of a cost model in which the cost for products depends on the time for production. The model has been chosen mainly due to its potential to represent different types of production models, i.e., (1) batch production, (2) continuous production, by setting the batch size to one and the batch setup time to zero, and (3) instant retrieval of products from storage, by setting the time-based production cost to zero, however without considering inventory costs for products.

3.1.2. Transportation

In TAPAS, transportation is carried out by a set of vehicles V , where a vehicle is characterized by its loading capacity (including type of storages), maximum traveling speed, fuel type, and emissions (e.g., NO_x , CO and CO_2) per consumed unit of fuel. It should be noted that we also denote electricity as a fuel type. The amount of fuel that a vehicle consumes depends on the weight of the carried load, and for vehicle $v \in V$, we let $fuel_v^{empty}$ denote the fuel consumption (per distance unit) when v is empty, and $fuel_v^{full}$ the fuel consumption when v is carrying its maximum load w_v^{max} . The fuel consumption $fuel_v(w)$ (per distance unit) when v carries a load of weight w ($0 \leq w \leq w_v^{max}$) is approximated linearly according to

$$fuel_v(w) = \frac{fuel_v^{full} - fuel_v^{empty}}{w_{vmax}} \cdot w + fuel_v^{empty}. \quad (3)$$

A vehicle has a transport mode (road, rail or sea) and it is only able to travel on links with the same mode. Further, it might be controlled by a timetable (offering so called timetabled transports with scheduled times for departure and arrival) or not (providing transports referred to as demand driven). Transport costs consist of:

- time-based costs (e.g., driver, capital, and administration),
- distance-based costs (e.g., fuel, vehicle wear, and kilometer tax),
- link-based costs (e.g., road tolls), and
- fixed operator-based ordering costs (e.g., administration).

3.1.3. Terminals

For terminals, the focus is on modeling times and costs for loading and unloading of vehicles. The times for loading and unloading a vehicle are expressed in terms of fixed and variable times. Fixed times $t_v^{loading\ fixed}$ and $t_v^{unloading\ fixed}$ represent the times it take to prepare a vehicle for loading or unloading. Variable times $t_{vp}^{loading\ variable}$ and $t_{vp}^{unloading\ variable}$ are given for each product type and denote the times needed for loading or unloading one unit of product p . The costs $c_v^{loading}$ and $c_v^{unloading}$ for loading and unloading a vehicle are given as costs per time unit. The cost for loading q units of product type p onto vehicle v is calculated as:

$$c_{vp}^{loading}(q) = c_v^{loading} \cdot t_{vp}^{loading}(q), \quad (4)$$

where the loading time is calculated as:

$$t_{vp}^{loading}(q) = t_v^{loading\ fixed} + t_{vp}^{loading\ variable} \cdot q. \quad (5)$$

The formulas for unloading cost and time are identical to (4) and (5), except for that the cost and times for loading are replaced with those for unloading. It is also possible to define vehicle specific fixed costs for visiting a terminal, e.g., to represent terminal fees, and terminal processing times in addition to the times for preparing a vehicle for loading and unloading, e.g., to be able to model customs.

3.2. The decision making simulator

In TAPAS we consider six transport chain roles (or decision makers); transport chain coordinator, product buyer, transport buyer, transport planner, production planner and customer. These roles can be argued to be present in all transport chains. However, a role can be associated with different logistical operators depending on the organizational settings within the chain. For instance, the transport chain coordinator may be represented by a producer in one transport chain and by a third party company in another.

In TAPAS, each customer node $n^c \in N^c$ is represented by a customer agent (C_{n^c}), each production node $n^f \in N^f$ by a production planner agent (\mathcal{P}_{n^f}), and transport planner agents ($\mathcal{TP} : s$) represent transport providers (operators). There exists one transport chain coordinator agent (\mathcal{TCC}), one transport buyer agent (\mathcal{TB}), and one product buyer agent (\mathcal{PB}). In a transport chain it is often possible to identify additional roles, such as terminal and inventory agents [27], and in TAPAS we have chosen to implicitly capture these roles in the physical simulator instead of modeling them as agents.

3.2.1. Interaction protocol for the ordering process

The process of buying resources (products or services) is a commonly occurring, and sometimes difficult, task, especially when the availability of resources are limited, different resources depend on each other, or multiple buyers and suppliers negotiate in parallel. The problem of how to automate the process of buying scarce resources is well covered in the literature (cf. [30,24,31]), but according to our knowledge there exists no efficient approach for dealing with resources that depend on each other. When buying resources that depend on each other, it is important that booking is made in a way that there will be no situations where one or more resources have been booked (and hence become useless) while it has become impossible to book some other resources that are required for accomplishing an overall task. Hence, synchronization and booking should be made in a way that it is guaranteed that either none or all required resources will be booked.

In TAPAS, a plan for fulfilling an order consists of a chain of tasks, starting with a production followed by a sequence of transports for different parts of the distance between the producer and the customer. The tasks in a plan need to be coordinated in a way that delivery is made inside the specified time window (if possible) and the starting and ending times of subsequent tasks are synchronized. If a resource included in a plan is no longer available at the time of booking, the whole plan may need to be reconsidered, and already booked resources may need to be canceled with a potential penalty as a consequence. A possible solution to this problem is to force resource providers to commit to their offers, as in the *contract net protocol* [35]. However, such an approach makes it more difficult to find good solutions, as well as obtaining high resource utilization in the system [24]. The reason is that resources may be allocated to potential buyers who later on decide not to

book them, at the same time as other potential buyers may need to make use of those particular resources in order to accomplish their goals.

In an approach for ordering matching products and transportation, two important challenges need to be considered: (1) products and transportation cannot be booked simultaneously, either products or transportation has to be booked first, and (2) products cannot be booked before finding a matching transport solution, and vice versa. In an earlier version of TAPAS [5], the contract net protocol was used for buying products from the supplier offering the best price, and a second contract net protocol was used to buy a matching transportation even though it can be considered hazardous to assume the availability of a cost efficient matching transport solution at the time of buying products.

In the current version of TAPAS we have dealt with this potential problem by applying a sequential ordering process, in which only one customer order is allowed to be processed at the same time. This guarantees that complete plans for obtaining products will be available until all included tasks have been booked. In the ordering process, which uses an interaction protocol originally presented in [21], *TCC* keeps a queue of incoming order requests that are processed in sequence. A similar approach, however applied to a manufacturing system, is presented in [25]. The obvious shortcoming of such an approach is that longer order processing times increase the risk that the order queue becomes longer and longer, especially when there is a high ordering frequency in the system, or if order processing times are long. In TAPAS, this is however not a problem, since order processing is assumed to be instantaneous.

In Fig. 2 we illustrate the interaction protocol that is used in TAPAS for fulfilling a customer order, and in Table 1 we describe the message types that are used in the interaction protocol. A new interaction starts when *TCC* receives an order request from a customer, and in two rounds of communication, a plan is first requested and generated (in the first ten messages) and then booked and confirmed (in the next ten messages). Hence it follows that customer agents are able to act proactively, while the other agent types are considered to be reactive. To improve the readability of the diagram, only the main flow of messages is shown. Also, the protocol can be interrupted in most of the steps, e.g., when *PB* fails to find an available supplier, or when *TB* is unable to find a feasible transportation for any product proposal.

3.2.2. Customer

Each customer node $n^c \in N^C$ is operated by a customer agent C_{n^c} who is responsible for ordering products in quantities that keep inventories at levels that minimize the costs for inventory holding and ordering, while reducing the risk for

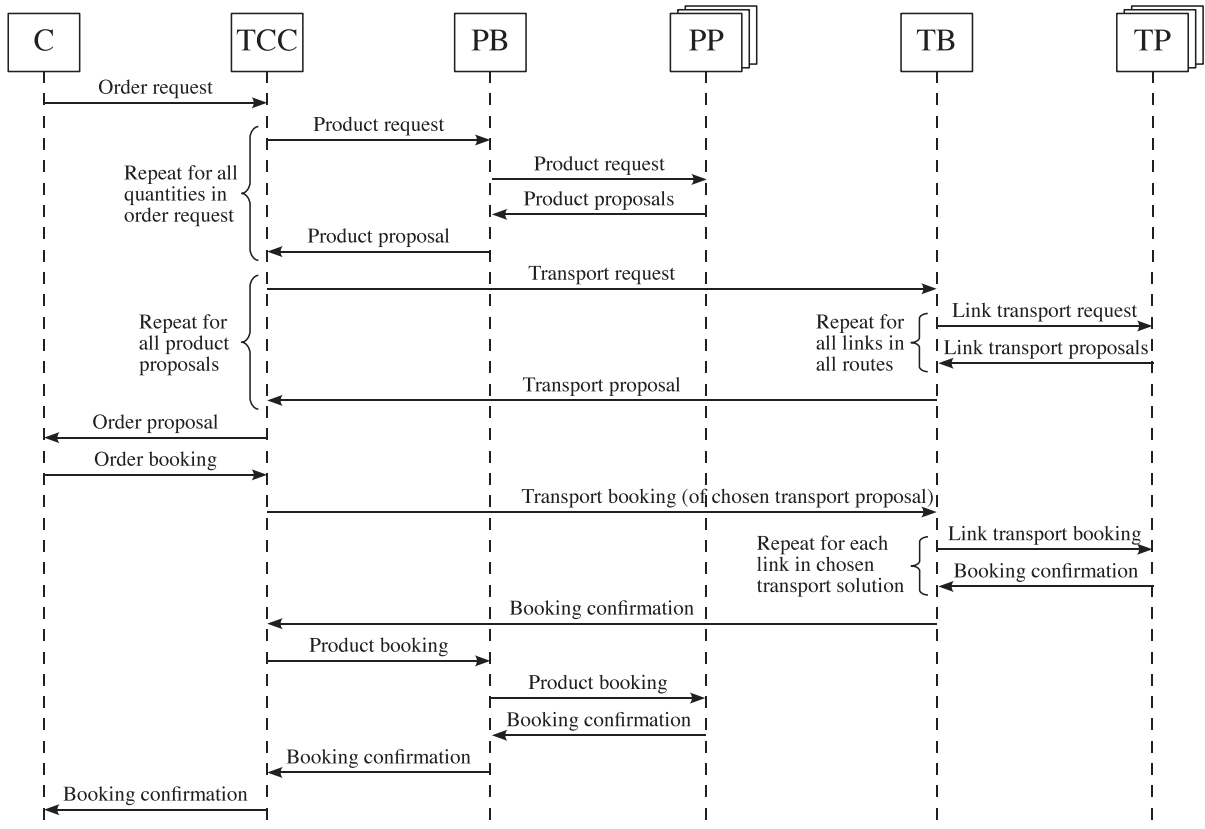


Fig. 2. Overview of the interaction protocol used in TAPAS only showing the main flow of messages.

Table 1

Specification of message types in order of appearance in the interaction protocol. The confirmation messages that are used in the protocol are considered self-explanatory, which is why we have chosen not to specify them.

Order request	Delivery node, product type, a set of candidate order quantities, and for each candidate quantity, a delivery time window, an upper limit for how much delivery is allowed to be delayed, and time-based convex penalty functions for delivering earlier or later than the window
Product request	Product type and quantity
Product proposal	Product type, quantity, pickup node, price, earliest time when the products can be picked-up, and supplier identifier
Transport request	Product type, quantity, pickup and delivery nodes, delivery time window, earliest time when pickup can be made, time-based convex delivery penalty functions, and an upper limit for delivery delay
Link transport request	Product type, quantity, departure and arrival nodes, and a time window in which transportation is considered
Link transport proposal (timetabled)	Price (differentiated on transportation, loading and unloading), route identifier, scheduled times for loading, unloading, departure and arrival, and transport operator identifier
Link transport proposal (demand driven)	Price (differentiated on transportation, loading and unloading), carrier type, availability interval, time needed for transportation, for loading and for unloading, and transport operator identifier
Transport proposal	Sequence of link transport proposals representing a transport solution and total price for the solution. For demand driven link transport proposals, times for departure, arrival, loading and unloading are specified. Link transport proposals in sequence are if possible re-represented (i.e., merged together) so that each link transport proposal has exactly one loading and exactly one unloading
Order proposal	Matching product and transport proposals
Order booking	Order proposal
Transport booking	Transport proposal
Link transport booking	Link transport proposal as specified in the transport proposal
Product booking	Product proposal and specification of pickup time

stockouts. When choosing an ordering quantity, a tradeoff has to be made between two counteracting costs; order cost (including costs for products, transportation, administration, etc.) and inventory holding cost. Typically, the order cost per unit decreases and the inventory holding cost increases with an increasing consignment size, and minimizing these costs gives the Economic Order Quantity (EOQ).

In the Wilson (EOQ) Model [46], which is claimed by Alstrom [2] to be the most commonly used policy for finding the EOQ, the annual costs for ordering (DC/Q) and inventory holding ($HQ/2$) is minimized by calculating EOQ according to $\sqrt{2CD/H}$, where C denotes the fixed order cost (independent on order quantity), D/Q the number of orders per year, H the annual holding cost, and $Q/2$ the average quantity in stock. However, the Wilson formula relies on a number of assumptions about the modeled system [3]. For instance, it assumes constant purchase price per ordered unit and constant order cost, which both are assumed to be independent on order quantity and time of order placement. This implies, e.g., that no volume discounts can be modeled. However, in TAPAS a customer always pays the current cost for products and transportation, and in particular the cost for transportation varies, e.g., due to load consolidation and departures of timetabled transports.

To be able to cope with non-constant order costs, we have used an approach based on an idea for dealing with deterministic time-varying demand [34], in which time is divided into disjoint time periods, and candidate order quantities are those that cover the demand for a number of future time periods. The quantity that minimizes the costs for ordering and inventory holding is then chosen. In TAPAS, a customer chooses the quantity that minimizes the cost for ordering and inventory holding among a set of predetermined candidate quantities. Hence, the choice of candidate order quantities is made differently but the evaluation of quantities is identical in the approaches.

At a particular point in time, the cost $c(p, q)$ per unit for ordering q units of product type p to node n^c is calculated as:

$$c(p, q) = \frac{c^{order}}{q} + \frac{c^{order} \cdot int_p \cdot t^{con}(p, q)}{2 \cdot q},$$

where c^{order} denotes the current order cost, i.e., costs for products and transportation, including time-based product cost during transportation, int_p the storage interest for product type p (in node n^c), and $t^{con}(p, q)$ the expected time for consuming q units of product type p .

To be able to determine when to place an order, it is usually assumed that demand is known and constant over time, and that the “order-to-delivery” lead time is fixed. In TAPAS, consumption is stochastic and lead times vary due to several factors, including choice of transport mode and timetabled departures. Hence, the lead time remains unknown until both product supplier and transportation have been chosen. The uncertainty of stochastic consumption is dealt with by modeling safety stock levels, and each customer uses an estimated (approximate) lead time, chosen as an estimated upper bound of the expected lead times for different transport alternatives and suppliers. Since demand is stochastic and lead-times are just estimations, a difficulty with the applied ordering strategy is how to determine when ordering should be considered. The traditional approach is to consider ordering as soon as the inventory level falls below a certain threshold level. This is typically implemented by checking the inventory level in intervals, which in our approach is referred to as *ordering opportunities*. The time between two subsequent ordering opportunities should be small enough to reduce the risk for stock-outs. Note that C_{n^c} (the customer agent) may have different ordering opportunities for different product types, and for each ordering opportunity a separate decision determines whether or not an order request should

be send. A decision whether or not to order at a particular point in time (i.e., an ordering opportunity) is based on the so called *order point*

$$\text{order}(p) = \text{saf}_p + \text{lead}_p \cdot \text{con}(p),$$

where saf_p denotes the safety stock level, lead_p the estimated “order-to-delivery” lead time, and $\text{con}(p)$ the forecasted consumption per time unit for product type p . If the current inventory level $\text{inv}^{\text{cur}}(p)$ plus the planned deliveries $\text{del}(p, \text{lead}_p)$ of product type p during the lead time period is less than $\text{order}(p)$, i.e.,

$$\text{inv}^{\text{cur}}(p, n^c) + \text{del}(p, \text{lead}_p) < \text{order}(p),$$

C_{n^c} asks for order proposals (in an order request) for a set of quantities Q .

A delivery time window should be chosen in a way that there will be enough available inventory capacity at the time of delivery, and that the risk for stock-out before delivery is kept at a minimum. For an order quantity of q units, the delivery time window is calculated as

$$\left[t^{\text{cur}} + a, t^{\text{cur}} + \max \left\{ a, \frac{\text{inv}^{\text{cur}}(p) + \text{del}(p, \text{lead}_p) - \text{saf}_p}{\text{con}(p)} \right\} \right],$$

where t^{cur} denotes the current time, $\text{inv}_p^{\text{max}}$ the maximum allowed inventory level for product p , and parameter

$$a = \max \left\{ 0, \frac{\text{inv}^{\text{cur}}(p) + \text{del}(p, \text{lead}_p) + q - \text{inv}_p^{\text{max}}}{\text{con}(p)} \right\},$$

makes sure that there will be enough available inventory capacity at the time of delivery.

3.2.3. Transport chain coordinator

\mathcal{TCC} is responsible for managing all customer orders, and received order requests are put in a queue for sequential processing. For each quantity $q \in Q$ in an order request, \mathcal{TCC} tries to find the least cost feasible combination of products and transportation, which are represented in an order proposal. The order proposals (one for each $q \in Q$) are returned to the customer who chooses the “best” proposal (i.e., the EOQ). To be able to generate order proposals, \mathcal{TCC} first asks \mathcal{PB} for relevant product proposals for each quantity $q \in Q$, by sending product requests. Then for each received product proposal, it asks \mathcal{TB} for a set of matching transport proposals, by sending transport requests.

3.2.4. Product buyer

\mathcal{PB} operates in between \mathcal{TCC} and the $\mathcal{PP} : s$, and it is mainly responsible for finding products that satisfy product requests that are received from \mathcal{TCC} . When receiving a product request \mathcal{PB} forwards it to all $\mathcal{PP} : s$, and the product proposals that are returned are forwarded to \mathcal{TCC} for further processing. When the customer has chosen an order proposal, which represents EOQ, \mathcal{PB} sends a product booking message to the chosen product supplier.

The behavior of \mathcal{PB} is not particularly advanced, and it can be argued that \mathcal{TCC} could communicate directly with the $\mathcal{PP} : s$ instead of communicating through \mathcal{PB} . We include \mathcal{PB} in the interaction protocol mainly for structural purposes (we argue that all transport chains should have a product buyer), and due to its potential for future improvement. An example of a possible refinement is to let \mathcal{PB} operate more proactively in the ordering process, e.g., by filtering out all product proposals except the best one in a particular geographic area.

3.2.5. Production planner

In TAPAS, each production node $n^f \in N^f$ is operated by a production planner agent \mathcal{PP}_{n^f} whose main responsibility is to plan the production in n^f . When \mathcal{PP}_{n^f} receives a product request asking for a product type $p \in P_{n^f}$ and quantity q , it generates a product proposal containing specification of the earliest time $t_{n^f}^{\text{prod}}$ when the requested products can be made available for pickup at n^f , and a price $c_{n^f}^{\text{prod price}}$, which is generated according to Eq. (2), however with the difference that the expected production time $t_{n^f}^{\text{prod}}(q)$ that is used as input to Eq. (2) is calculated in Eq. (1) assuming that a batch setup time is considered for all batches.

A production cost that is calculated using the model described in Section 3.1.1 depends on whether or not there need to be batch setups involved in the production, and in particular it depends on the batch sequencing, which typically varies according to the changing demand for products. It is therefore not possible to use Eq. (1) to generate the production time when using the production cost model as a product price model without assuming that, e.g., a batch setup time is included in each produced batch. Otherwise, there would potentially be unrealistic dependencies between the production cost, which depends on factors such as batch sequencing, and the price for products, which normally is rather static. In summary, the production cost model used in TAPAS assumes that setup is considered only when there is a change in product type, and the product price model assumes that setup time is considered in all batches. Hence, the product price is (1) an

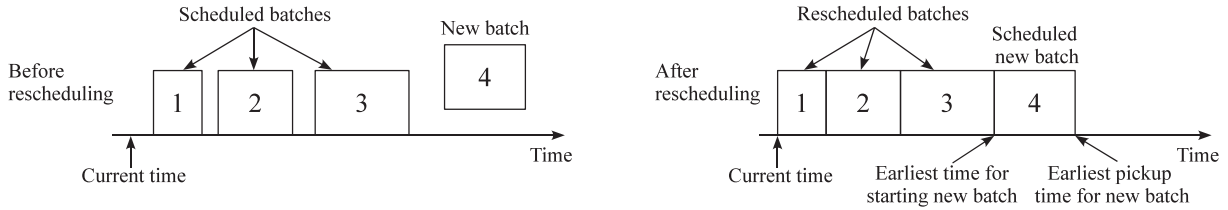


Fig. 3. Illustration of how three scheduled batches (1–3) can be rescheduled in order to enable early processing of an incoming order (4).

upper bound of the production cost, and (2) it does not depend on batch sequencing, which is important in a model like TAPAS.

To enable an incoming order to be processed as early as possible, it is assumed that the existing production plan can be rescheduled so that already scheduled products are produced earlier than originally planned, assuming there is available production capacity for rescheduling. We have chosen not to consider the additional inventory costs that might be a consequence of rescheduled production, since that would considerably increase the modeling complexity. In Fig. 3 we illustrate how a production plan can be updated to enable early processing of an incoming order.

The rescheduling principle illustrated in Fig. 3 is used by \mathcal{PP} :s both when generating product proposals and when managing product bookings, i.e., when receiving product booking messages. When \mathcal{PP}_{nf} receives a product request, it calculates the earliest possible pickup time t_{nf}^{prod} by checking the earliest time that all scheduled productions can be finished. This is achieved by imagining a sequence of all scheduled productions (starting from the time of the request) in a way that there are no gaps in between any productions.

When receiving a product booking message, \mathcal{PP}_{nf} first checks if there is enough available capacity for production immediately before the pickup time that is specified in the product booking message. In other words, it checks whether or not the production can be planned without rescheduling. If that is the case it schedules the production so that the booked products is expected to be available immediately before the requested pickup time. It should be noted that such a procedure typically creates gaps in the production plan, as illustrated to the left in Fig. 3. If there is insufficient capacity for production immediately before the specified pickup time, \mathcal{PP}_{nf} has to reschedule the existing production plan so that the requested products can be produced immediately before the pickup time. The reason that productions are always scheduled as late as possible it that it is believed that such a policy in most cases creates a minimum need for rescheduling. The rescheduling procedure can be seen as recursive, since moving a scheduled production earlier in time might create a need to reschedule the production immediately before that production, which might cause a need to reschedule the production even before that, etc. It should be noted that it is assumed that the pickup time specified in a product booking message never violates the earliest pickup time given in the corresponding product proposal.

3.2.6. Transport buyer

In TAPAS, the transport buyer (\mathcal{TB}) is responsible for creating transport solutions in order to satisfy transport requests received from \mathcal{TCC} . A transport solution is obtained by solving a shortest (cheapest) path problem with additional constraints for representing the following complicating factors:

1. Vehicle capacities and previous bookings must be considered, e.g., the remaining capacity on a booked vehicle should be made available for future bookings.
2. There is a mix of timetabled and demand driven transports.
3. Terminal activities such as unloading, loading, and reloading should be explicitly considered.
4. Delivery has to occur inside the delivery time window, otherwise a penalty cost should be applied.
5. Time-based, link-based, distance-based and fixed transport costs (as discussed in Section 3.1.2), and time-based costs, e.g., for tied up capital and deterioration should be considered.

The literature contains a number of approaches to intermodal freight planning (e.g., [8,50,9,23]). However, none of them applies to our problem since either (1) they do not handle all complicating factors that we need to consider, or (2) the combination of timetabled and demand driven transports are handled in an unsatisfactory way, e.g., by representing a demand driven transport as multiple timetabled departures, i.e., one for each time period. Such a modeling approach may have severe effects on the performance of the algorithm if the number of time-periods is high. Therefore, we have developed a customized algorithm for finding transport solutions. A compressed pseudo code description of the proposed algorithm is given in Algorithm 1. In the algorithm, a set of precompiled (manually specified) routes $R_{n_f n_c}$, expressed as sequences of network nodes is defined for each potential pair of pickup and delivery nodes $n_f \in N^F$ and $n_c \in N^C$. Since routes are defined as node sequences, two sequenced nodes in a route may be connected by multiple links, e.g., corresponding to different transport modes. The motivation for using precompiled routes is that useful routes generally are known or can be found in advance.

Algorithm 1. Find the least cost transport solution for a transport request, given a pickup node n_f , a delivery node n_c , and link transport proposals for all connections in the routes in $R_{n_f n_c}$. The cost for a tree node corresponds to the cost for traveling from the departure node of the link transport proposal it represents until unloading at n_c .

```

procedure FINDBESTTRANSPORTSOLUTION
  create rootNode ▷ rootNode represents the delivery time window
  initialization: rootNode.cost := 0, rootNode.bestChild := null, bestCost := ∞
  for each route  $r \in R_{n_f n_c}$  do
    for each link transport proposal  $t$  in  $r$  which has arrival node  $n_c$  do
      newChild = createChild(rootNode,  $r$ ,  $t$ , bestCost)
      if newChild.cost < rootNode.bestCost then
        rootNode.bestChild := newChild, bestCost := newChild.cost
  return best transport solution with cost bestCost ▷ the best solution is obtained from the best child
procedure CREATECHILD(parentNode,  $r$ ,  $t$ , bestCost)
  create new node currentNode
  initialization: currentNode.bestChild := null, currentNode.bestChildCost := ∞
  if  $t$  is infeasible in combination with the transport represented by parentNode then
    currentNode.cost := ∞, return currentNode
  if  $t$  is demand driven then
    set preliminary times for departure and arrival, and potentially for loading and unloading *
    calculate currentNode.cost ▷ includes cost for transportation, loading, unloading, penalty, etc.
    currentNode.cost = currentNode.cost + parentNode.cost
    if  $t$ .departureNode =  $n_f$  then return currentNode.cost ▷ a complete transport solution is found
    if currentNode.cost ≥ bestCost then currentNode.cost := ∞ return currentNode
    for each link transport proposal  $t'$  in  $r$  that is connected to the arrival node of  $t$  do
      newChild = createChild (currentNode,  $r$ ,  $t'$ , bestCost)
      if newChild.cost < currentNode.bestChildCost then
        currentNode.bestChild := newChild, currentNode.bestChildCost := newChild.cost
  return currentNode

```

* Potentially involving a backtracking towards rootNode in order to represent updated preliminary times for a sequence of demand driven ancestor nodes.

In the algorithm, \mathcal{TB} processes the routes $r \in R_{n_f n_c}$ in sequence to be able to find the least cost transport solution, possibly including penalty cost, for which a transport proposal is created and returned to \mathcal{TC} . The algorithm is applied when \mathcal{TB} has received all link transport proposals for each pair of sequenced nodes (n_i, n_j) included in any route $r \in R_{n_f n_c}$ (see the interaction protocol in Fig. 2). The link transport proposals that are returned from the \mathcal{TP} :s should have departure and arrival times, or intervals of availability for demand driven transports, inside an interval $[t_{n_f}^{prod}, t_{n_c}^{late} + k^{late}]$, where $t_{n_f}^{prod}$ is the earliest time the products can be available for pickup at n_f , $t_{n_c}^{late}$ is the end of the delivery time window $[t_{n_c}^{early}, t_{n_c}^{late}]$, and $k^{late} \geq 0$ is a constant that is used to allow delivery to be made later than the delivery time window. However, penalty costs as specified by the customer are applied to deliveries earlier or later than the delivery time window.

In the search for the optimal transport solution the algorithm uses a search tree T , in which each tree node η (except for the root of the tree) represents a link transport proposal with departure (network) node $dep(\eta)$ and arrival node $arr(\eta)$. Instead of representing a transport proposal, the root ρ represents the delivery time window and serves as the parent of all tree nodes representing transports arriving at n_c . The search starts in n_c and proceeds backwards (along the routes $r \in R_{n_f n_c}$) towards n_f by gradually expanding the search tree. For a search tree node η , one child node will be added for each link transport proposal representing a transport arriving at $dep(\eta)$. Hence, it follows that a link transport proposal will be represented by multiple tree nodes in order to enable evaluation of each possible combination of link proposals along the routes. Each search tree node η has a cumulative cost $c(\eta)$, which corresponds to the cost for moving the goods from $dep(\eta)$ to n_c (including unloading at n_c). In the algorithm the following search strategies are applied:

1. Tree nodes for which the transports they represent are infeasible in combination with the transports represented by their ancestor nodes, as well as nodes that necessarily lead to transport solution that are more expensive than the so far best found solution will be pruned (i.e., no child nodes will be added).
2. A depth-first based traversal strategy is adopted in order to potentially find valid solutions early to enable early pruning of more expensive tree branches.
3. Tree nodes representing timetabled link transport proposals are expanded before nodes representing demand driven link transport proposals.

The main challenge for the algorithm is to deal with the differences between timetabled and demand driven transports, where a previously booked demand driven transport is regarded as timetabled. The difficulty is that timetabled transports have fixed departure times while demand driven transports are flexible concerning departure and arrival times. For simplicity of presentation, we here refer to tree nodes representing timetabled and demand driven transports as timetabled and demand driven nodes.

When expanding a search tree node, the feasibility is checked for all added child nodes, however in different ways for timetabled and demand driven nodes, as explained below. Based on the departure and arrival times of its ancestors, for a demand driven node the algorithm specifies preliminary times for departure, arrival and potentially for loading and unloading. For a demand driven node it also specifies a possible *interval of operation*, defining when the activities for the represented transport are allowed to take place. For a demand driven transport, the interval of operation is based on the interval of availability that is specified in the link transport proposal, as well as on the operation times of the ancestors, and it may need to be used further down in the tree, e.g., when a timetabled successor requires that the preliminary specified times for one or more demand driven ancestors should be updated. For a parent node η and child node ζ , we let $z(\zeta, \eta) \in \{0, 1\}$ denote whether or not the products will be reloaded ($z(\zeta, \eta) = 1$) or not ($z(\zeta, \eta) = 0$) between the transports represented by ζ and η , and $c^{\text{reloading}}(\zeta, \eta) = c^{\text{unloading}}(\zeta) + c^{\text{loading}}(\eta)$ the cost for reloading. Since it is only known by the child whether or not the goods need to be reloaded, the cost for loading has to be represented in the child node. Moreover, since a parent typically has multiple child nodes, a reconsidering of the preliminary times for a demand driven node needs to be represented in the particular successor node that requires that the times should be updated.

The root. As mentioned above, the root node ρ represents the delivery time window, and a timetabled transport arriving at n_c (i.e., a timetabled child of ρ) will always be valid since a penalty cost can be applied if delivery is made outside $[t_{n_c}^{\text{early}}, t_{n_c}^{\text{late}}]$. Similarly, a demand driven transport is valid whenever the time interval for when it is available, as specified in the link transport proposal, is at least as long as the time it takes for the vehicle to perform its activities. For a demand driven transport, preliminary times for unloading, arrival and departure are set as late as possible in the delivery time window, and if delivery inside the delivery time window is impossible due to the interval of availability, the preliminary times are heuristically chosen either before or after the delivery time window in a way that the penalty is minimized. The general form for describing the cost for a node that connects to ρ is:

$$c(\zeta) = c^{\text{transp}}(\zeta) + c^{\text{unloading}}(\zeta) + c^{\text{time}}(\zeta) + \text{pen}(\zeta),$$

where $c^{\text{transp}}(\zeta)$ is the transport cost, $c^{\text{unloading}}(\zeta)$ the unloading cost, $\text{pen}(\zeta)$ the potential penalty cost that need to be considered when adding ζ , and $c^{\text{time}}(\zeta)$ a time-based product cost, i.e., the mathematical product of the product value, the time between departure and unloading, and a time-based factor (or interest rate). The customer chooses this time-based factor, and it may represent a cost for tied-up capital, deterioration, time to market, etc., for the particular type of product.

Intermediary nodes. When adding a child node ζ to an intermediary search tree node η (i.e., η is not the root), four cases can occur depending on what types of transports η and ζ represent. It should be noted that the transport represented by ζ occurs before the transport represented by η . With the notation introduced above, the cost $c(\zeta)$ for reaching ζ is described as:

$$c(\zeta) = c(\eta) + c^{\text{transp}}(\zeta) + z(\zeta, \eta) \cdot c^{\text{reloading}}(\eta, \zeta) + c^{\text{time}}(\zeta) + \text{pen}(\zeta),$$

where in particular $\text{pen}(\zeta)$ and $c^{\text{time}}(\zeta)$ depends on several factors. The base case is that a time-based cost $c^{\text{time}}(\zeta)$ should be generated for the time between the departure times of ζ and η .

If both η and ζ are timetabled, it is only necessary to conduct a feasibility check to ensure that ζ will be further expanded only if the arrival time of ζ is earlier than the departure time of η , and if η and ζ represent transports by different vehicles there also need to be time for reloading.

When η is timetabled and ζ is demand driven, the time for departure and loading in node η are strictly specified in the corresponding link transport proposal, and the times for the activities represented by ζ should be synchronized with the activities in node η . Since η represents a timetabled vehicle while ζ represents a demand driven one (a vehicle is assumed to offer either timetabled or demand driven transports), it is assumed that reloading will occur between ζ and η . The unloading of ζ will be preliminary scheduled immediately before η starts loading, and the actual transport (departure and arrival) immediately before that. If ζ is unavailable at the preferred time of operation, instead the activities in ζ should be preliminary scheduled earlier in the period of availability in a way that (as late as possible) the penalty is minimized.

The case when both η and ζ are demand driven is similar to the previous case. However, if reloading needs to occur between η and ζ (i.e., η and ζ represent different vehicle types), the loading of η needs to be preliminary scheduled, as this was impossible to do when η was created. However, the preferred time for loading in η might be infeasible due to the interval of operation for η , which means that the preliminary scheduled activities for η will have to be rescheduled (actually delayed) and additional penalties, as well as an additional time-based cost ($c^{\text{time}}(\zeta)$) should be considered. Delaying η means that additional demand driven nodes, i.e., ancestors of η , also may have to be delayed, depending on how their activities are preliminary planned. Actually, the preliminary times of all sequential demand driven nodes that can be found in a back-track search (towards the root ρ) until either a timetabled node or the root is reached, may need to be rescheduled. If (in the back-track

search) a timetabled node ψ is reached before reaching ρ , the preliminary times for the activities represented by the demand driven ancestors between ζ and ψ should be synchronized also considering the loading time of ψ (note that reloading will occur since ψ is timetabled and its child in the current branch is demand driven). If ρ is reached before any timetabled node could be found, i.e., ρ and ζ is connected by a sequence of demand driven transports, a penalty for delivering outside the specified delivery time window may have to be applied as a possible consequence of rescheduling the node connected to ρ . Another reason for rescheduling a sequence of demand driven ancestors is a limited availability interval of ζ . Of course, limited intervals of operation for one or more of the demand driven transports that need to be rescheduled may limit how rescheduling may be conducted, and in worst case ζ will be infeasible, and hence pruned. Since each of the, potentially rescheduled, ancestors may have multiple successors, it is important to emphasize that extra penalty costs and time-based costs that are a consequence of the rescheduling of ancestors have to be represented in ζ . Also, since a rescheduled node may have other branches, which require different or no rescheduling, a specification of how activities for the ancestors are rescheduled should be represented in ζ .

The case when η is demand driven and ζ is timetabled is similar to the case when both η and ζ are demand driven. The difference is that for ζ , the times for loading, unloading, and traveling are represented in the corresponding link transport proposal. As in the previous case, it might be necessary to reschedule η and a sequence of demand driven ancestors of η whenever the times of ζ causes infeasibility with the preliminary times of η . In the same way as discussed above, penalty cost and an additional time-based cost may have to be considered.

Leaf nodes. Leaf nodes, which represent transports departing from n_f , are processed exactly as intermediary nodes, with the difference that loading also need to be scheduled for ζ . This might lead to infeasibility and extra penalty cost and time-based cost, in the same way as discussed above. In general, the cost for a leaf node can be calculated as:

$$c(\zeta) = c(\eta) + c^{\text{transp}}(\zeta) + z(\zeta, \eta) \cdot c^{\text{reloading}}(\zeta, \eta) + c^{\text{loading}}(\zeta) + c^{\text{time}}(\zeta) + \text{pen}(\zeta).$$

When the best sequence of tree nodes (a transport solution) for a transport request has been determined, the preliminary times for all demand driven transports in the solution need to be permanently set. This may require that demand driven transports included in the solution needs to be rescheduled according to the preferences of nodes further down in the tree (i.e., closer to the leaf). If more than one node specifies a rescheduling of the same demand driven ancestor, it is the preferences of the node closest to the leaf that should be applied. The reason is that the preferences of a node that is added later in the tree also takes into account the preferences of nodes that have been added earlier.

3.2.7. Transport planner

A transport planner agent (\mathcal{TP}) represents an owner of a fleet of vehicles $V_T \subseteq V$, which are assumed to operate in some particular geographic area, and the main responsibility of \mathcal{TP} is to plan the vehicles in V_T by assigning them transport tasks. When \mathcal{TP} receives a link transport request from \mathcal{TB} , specifying start node n_s , end node n_e , and a window $[t^s, t^e]$, it creates a set of link transport proposals that are returned to \mathcal{TB} . Since timetabled vehicles depart and arrive according to timetables, while demand driven vehicles are flexible regarding when they are able to operate, it follows that link transport proposals for the two different types of transports need to contain somewhat different information. For a timetabled transport, a link transport proposal specifies the scheduled times for loading, unloading and transportation, and for a demand driven transport, instead the time needed for traveling, loading and unloading, as well as a time interval in which the transport is available for operation is specified. For a timetabled vehicle, a separate link transport proposal will be generated for each scheduled departure between n_s and n_e , which has departure and arrival inside $[t^s, t^e]$. For demand driven transports, a separate link transport proposal can be generated for each available demand driven vehicle, but since this may result in a large number of similar (or identical) link transport proposals, instead one proposal, specifying an interval of availability is generated for each different vehicle type in V_T . Some advantages with this approach are that the interval of availability for a vehicle type may be considerably larger than those for the individual vehicles of the particular type, and the number of link transport proposals can be reduced. Moreover, link transport proposals for previously booked demand driven transports are generated as for timetabled transports, and each scheduled demand driven departure (from n_s to n_e within the requested time interval) with remaining capacity should be represented in a separate link transport proposal, which specifies the planned times for departure, arrival, loading, and unloading.

Typically, a vehicle carries products for more than one customer (i.e., load consolidation), and a buyer of transport capacity should pay a reasonable share of the total transport cost. However, when giving a link transport proposal, \mathcal{TP} only knows about already booked consignments. Future bookings can only be guessed, or estimated from forecasts. Also, how much goods will be carried on the return transport is something that potentially could have an effect on the price for transport capacity. We do not model return transports explicitly, but return transports can implicitly be represented by adjusting different cost components and the average load utilization factor, and it is the responsibility of the user to decide if and how return transports should be considered. In summary, the major difficulty for \mathcal{TP} is to determine prices for those transport bookings that do not utilize the full capacity of a vehicle.

An obvious approach for setting the transport price is to charge a price that is based on the weight and size of the goods to be transported, as well as on an average load utilization factor. In this approach, no volume discounts are modeled, and the cost for buying transport capacity is calculated as:

$$c^{order} = \frac{c^T \cdot f^L}{f^{avg}} + c^{fuel}(w), \quad (6)$$

where c^T is the total transport cost (including fuel cost for an empty transport), f^L the load utilization factor for the requested load, f^{avg} the assumed average vehicle load utilization factor (potentially different for different vehicle types, geographical areas and transport operators), and $c^{fuel}(w)$ the fuel addition cost based on the weight w of the requested load. The load factor (f^L) is calculated as the requested quantity q divided by the maximum loading capacity of the requested product type, restricted either by the volume or weight capacity of the vehicle. As mentioned above, the fuel cost for a transport is included in c^T , and the fuel addition cost $c^{fuel}(w)$ is calculated as:

$$c^{fuel}(w) = l \cdot (fuel_v(w) - fuel_v^{empty}), \quad (7)$$

where l is the traveled distance and $fuel_v(w)$, which is the fuel consumption per distance unit when carrying a load of weight w , is calculated according to Eq. (3).

In addition to the linear pricing model described above, we provide a pricing model in which a risk cost is added to bookings that leave transport capacity unbooked. The risk cost is based on an estimated probability d , at the time of giving a link transport proposal, that there will be at least one more booking before the departure. Typically, the risk costs for bookings decrease as the amount of booked transport capacity increases, and it is the responsibility of the user of TAPAS to specify how the probabilities should be calculated.

In the pricing model with a risk cost, the order cost is calculated as:

$$c^{order} = c^T \cdot f^L + c^{fuel}(w) + risk, \quad (8)$$

where the risk cost is calculated as:

$$risk = (1 - d) \cdot c^T \cdot f^R, \quad (9)$$

and f^R is the remaining transport capacity, at the time of booking, in percentage of the total capacity.

In summary, we propose two different approaches for setting the price for transport capacity; one where the price depends linearly on how much capacity is ordered, and one where early bookings gets a larger share of the transport cost to cover for the uncertainty regarding future bookings.

3.3. Implementation

TAPAS is implemented as a discrete event simulator, since we found it appropriate to model the studied domain as a chronological sequences of events, representing activities such as order processing, loading, unloading, departures, and arrivals. It makes use of a next-event time advance mechanism, instead of fixed-increment time advance approach, due to its ability to simulate time in an arbitrary level of detail. TAPAS is implemented in the Java language, and the decision making simulator is implemented using the Java Agent DEvelopment Framework (JADE) platform [4].

Agents operate autonomously and typically asynchronously, and in agent-based (parallel) simulation it is important to carefully consider how to synchronize activities with the simulation clock. There are two approaches for synchronizing parallel simulation (cf. [48]); in *conservative protocols* activities are always simulated in chronological order, and in *optimistic protocols* activities are allowed to be simulated in non-chronological order, however with the consequence that the simulation needs to be recovered when synchronization errors occur. TAPAS makes use of a conservative approach, in which a *synchronization agent* is used to ensure that activities always are simulated in chronological order. The synchronization agent manages the simulation clock and it maintains the *event queue*, in which future events are sorted in chronological order. It should be noted that, at a particular point of time only some future events are typically known; events are continuously generated as the simulation proceeds over time. The behavior of the synchronization agent can be described as a loop in which it (1) removes the first event (i.e., next in time) from the event queue and (2) advanced the simulation clock to the time of the event that was removed from the queue. It also sorts new events into the event queue whenever they are created.

Activities are in TAPAS represented by start and end events, which makes it straightforward to simulate that activities may take longer time than expected, i.e., delays in activities. In TAPAS it is possible to model production and transportation using user-defined probability distributions. Start events are created as consequences of other activities, e.g., since order processing occurs at fixed points in time the start event for next order processing is created when processing the end event for the previous order processing, and events for starting production and transportation are generated at the time of booking those activities. The end event for an activity is created at the start of the activity (i.e., when the start event is processed). The delay for an activity is known already when the activity starts, and it is therefore captured in the time for the end event of the activity.

Those physical entities that perform activities (not entities such as links and nodes) are in TAPAS connected to agents. For example, vehicles are connected to transport planners and production facilities to production planners. When the synchronization agent processes an event, it sends a message to the agent connected to that event, providing informing about the activity that should start or end (depending on whether the processed event is a start or end event). In case the activity cannot start (or end) at the particular point of time, the agent returns a message informing about that. The synchronization

agent is equipped with a mechanism for handling delays and dependencies between activities. In addition to managing the event queue, the synchronization agent also keeps a list of events that for different reasons are unable to start, e.g., since they are waiting for other, potentially delayed, activities to end. For example, the loading of a vehicle in a producer depot may have to wait for a delayed production to finish. Whenever an activity has ended (i.e., an end event has been processed) the synchronization checks if any activity in the waiting list could start as a consequence of the ended activity. If that is the case, the synchronization agent creates a start event for the activity that might start and schedules it in the event queue as early as possible.

4. Scenario and simulation study

In this section we illustrate the functionality included in TAPAS by briefly describing a simulation study that has been conducted in a scenario around the Southern Baltic Sea. The scenario is an extension of a scenario that has been studied in collaboration with partners in an EU financed project called EastWest Transport Corridor (<http://www.eastwesttc.org>). It contains one logistical terminal, in Kaunas (Lithuania), which provides two types of products, and three customers; one in Sweden (Älmhult) and two in Denmark (Copenhagen and Esbjerg). In the scenario, which is illustrated in Fig. 4, transportation of 20 ft. ISO-containers (TEUs) from Kaunas to the three customers is offered by five transport providers. Transportation by sea and rail is assumed to follow timetables while transportation by road is demand driven. A detailed description of the scenario, including input parameters for all entities, timetables, etc., is given in [20].

We studied the consequences of three transport policy and infrastructure measures aimed at achieving a modal shift from road to rail and sea transportation, which is a goal within the European Union [16]; an introduction of a kilometer tax for heavy trucks in Sweden and parts of Denmark, a CO₂ tax for all types of transports in the studied area, and a new direct railway link between Karlshamn and Älmhult (the so-called SouthEast Link). The kilometer tax level is suggested in [17] and it is differentiated based on the euro class, as well as on the total weight of the trucks, the CO₂ tax levels are similar to the levels discussed in [33], and the SouthEast Link is an infrastructure project that is currently discussed in Sweden. In the study, we investigated two different timetables for the SouthEast link.

The scenario and its results have been validated through interviews with domain experts, and a visualization of the scenario helped us discover unrealistic assumptions and to facilitate the communication of assumptions and simulation results. We have performed sensitivity analyses of different input parameters in order to understand what influence different parameters have on the results. Moreover, the estimated transport cost structures, i.e., the relations between time-based and distance-based costs have been compared to the cost structures used in the Samgods model [38].

In the simulation study we considered the following experimental settings:

- S0.** The base case refers to the current situation in which none of the studied measures are applied.
- S1.** S0 + a kilometer tax of 0.15 euro/km for trucks operating in Sweden, and between Copenhagen Terminal and Copenhagen Customer.
- S2.** S0 + a CO₂ tax for all vehicles operating in the modeled region. Tax levels from 0.10 euro/kg up to 0.30 euro/kg in steps of 0.05 were considered. We let S2.x refer to setting S2 with a CO₂ tax level of 0.x euro/kg.
- S3.** S0 + a new railway link between Karlshamn and Älmhult (i.e., the SouthEast Link). Two timetables, which are synchronized in different ways with ferry arrivals in Karlshamn (from Klaipeda), were considered: (a) worse synchronization and (b) better synchronization (see [20] for specification of timetables).

For each setting we simulated 420 days with a precision of 1 min. We made simulation runs with 10 sets of random generator seeds for variation of consumption (different seeds were used for different customers). Each set of seeds was used for all settings, which enabled us to make pair-wise comparisons of results for different settings.

From a larger set of available routes (see [20] for specification of all routes available in the scenario), we observed that the following five routes were used, however in different proportions for different settings:

- Route 1.** Kaunas (Rail) Klaipeda (Sea) Karlshamn (Road) Älmhult
- Route 2.** Kaunas (Rail) Klaipeda (Sea) Karlshamn (Rail) Älmhult
- Route 3.** Kaunas (Rail) Klaipeda (Sea) Karlshamn (Rail) Copenhagen Terminal (Road) Copenhagen Customer
- Route 4.** Kaunas (Rail) Klaipeda (Sea) Karlshamn (Road) Copenhagen Customer
- Route 5.** Kaunas (Rail) Klaipeda (Sea) Fredericia (Road) Esbjerg

An important indicator of the impact of the studied measures is the route choice, which we illustrate with the percentage of TEUs transported using different routes. In Table 2 it can be seen that all of the studied measures caused a shift towards routes involving more rail transports and less road transports. However, for different measures the shift was observed in different parts of the network. For transportation to Älmhult, the only measure that showed an effect on the route choice is the SouthEast Link. In the settings without the SouthEast Link, all TEUs were transported on road between Karlshamn and Älmhult (i.e., Route 1). In the settings with the SouthEast Link (S3a and S3b) we observed shifts toward Route 2 using the SouthEast Link (i.e., railway). In S3a in average 5.8% and in S3b in average 43.6% of the TEUs were transported using Route 2. Not surprisingly, in the setting with better timetable synchronization (S3b) we observed a higher shift than in the setting with

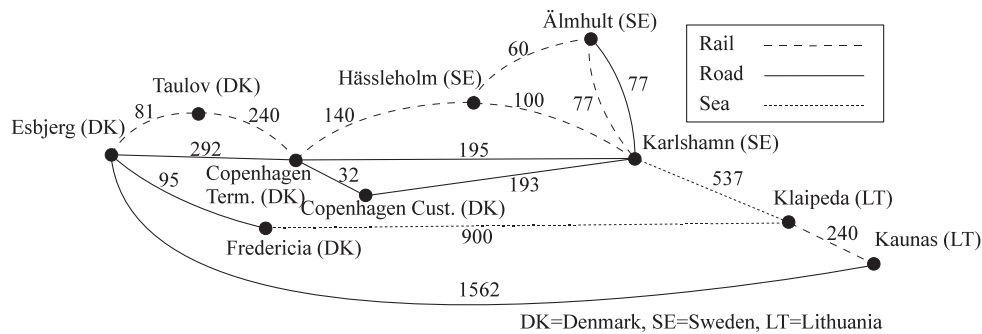


Fig. 4. Illustration of the transport network used in the scenario, where the numbers on the links represent distances in kilometers.

slightly worse synchronization (S3a). In all settings, transportation of all TEUs to Esbjerg were made using Route 5, with sea transportation from Klaipeda to Fredericia followed by road transportation from Fredericia to Esbjerg. This is reasonable due to the fact that the long distance makes it economically tractable to use sea transportation instead of land transportation through Sweden and Denmark. For Copenhagen Customer, which can be reached only by truck, the results vary for the different settings. For settings S0 and S3, in which no measures were applied for transportation on the routes to Copenhagen Customer, all transports were made using road transportation from Karlshamn directly to Copenhagen Customer (i.e., Route 4). In setting S1 (kilometer tax), a 100% shift towards Route 3 using rail between Karlshamn and Copenhagen Terminal followed by road transportation between Copenhagen Terminal and Copenhagen Customer was observed. For settings S2 (CO₂ tax), a gradually increasing shift towards Route 3 was observed as the tax level was increased.

A positive consequence of achieving a shift from road to rail transportation is a reduced amount of CO₂ emissions. All observed reductions of CO₂ emissions in the studied system was a consequence of a modal shift from road to rail in Sweden and Denmark. In Table 2 we show the average CO₂ emissions (in tons), as well as the average transport work (in tonne-kilometer), which is another important indicator of a changed transport pattern, for transportation of one TEU from Kaunas to each customer. Moreover, in Fig. 5 we present the relative CO₂ reduction for (a) the whole system, and (b) land transports in Sweden and Denmark. The main reason for observing a small reduction of CO₂ emissions when considering all simulated transports is that a significant share of the transports in all settings were made using sea and rail transportation between Kaunas and Karlshamn and between Klaipeda and Esbjerg. Further, the transport costs are affected in different ways by different measures and it may be important to analyze the positive effects (e.g., reduced CO₂ emissions) in relation to the economic impact caused by applying measures. For the studied measures, we show in Table 2 the average costs for transporting one TEU to the different customers.

Table 2

For each setting, the average taken over 10 replications for a number of relevant types of output data.

		S0	S1	S2.10	S2.15	S2.20	S2.25	S2.30	S3a	S3b
<i>Share of TEUs (in %) transported using different routes (differentiated on customer)</i>										
Älmhult	Route 1	100	100	100	100	100	100	100	94.2	56.4
	Route 2	0	0	0	0	0	0	0	5.8	43.6
Copenhagen	Route 3	0	100	49.6	49.6	66.0	78.5	100	0	0
	Route 4	100	0	50.4	50.4	34.0	21.5	0	100	100
Esbjerg	Route 5	100	100	100	100	100	100	100	100	100
<i>Average transport work (in tonne-kilometer) for transportation of one TEU to each customer (differentiated on transport mode)</i>										
Älmhult	Rail	2640	2640	2640	2640	2640	2640	2640	2689	3009
	Road	847	847	847	847	847	847	847	798	478
	Sea	5907	5907	5907	5907	5907	5907	5907	5907	5907
Copenhagen	Rail	2640	5280	3949	3949	4382	4713	5280	2640	2640
	Road	2123	352	1245	1245	954	733	352	2123	2123
	Sea	5907	5907	5907	5907	5907	5907	5907	5907	5907
Esbjerg	Rail	2640	2640	2640	2640	2640	2640	2640	2640	2640
	Road	1045	1045	1045	1045	1045	1045	1045	1045	1045
	Sea	9900	9900	9900	9900	9900	9900	9900	9900	9900
<i>Average CO₂ emissions (in tons) for transportation of one TEU from Kaunas to each customer</i>										
Älmhult		0.623	0.623	0.623	0.623	0.623	0.623	0.623	0.622	0.616
Copenhagen		0.679	0.660	0.670	0.670	0.667	0.664	0.660	0.679	0.679
Esbjerg		0.885	0.885	0.885	0.885	0.885	0.885	0.885	0.885	0.885
<i>Average cost (in euro) for transporting one TEU to each customer</i>										
Älmhult		364.2	368.5	426.0	456.9	487.8	518.7	549.6	364.7	364.3
Copenhagen		554.8	566.3	627.3	660.9	695.4	729.4	764.5	554.9	554.8
Esbjerg		519.0	519.0	606.9	650.8	694.7	738.6	782.6	519.0	519.0

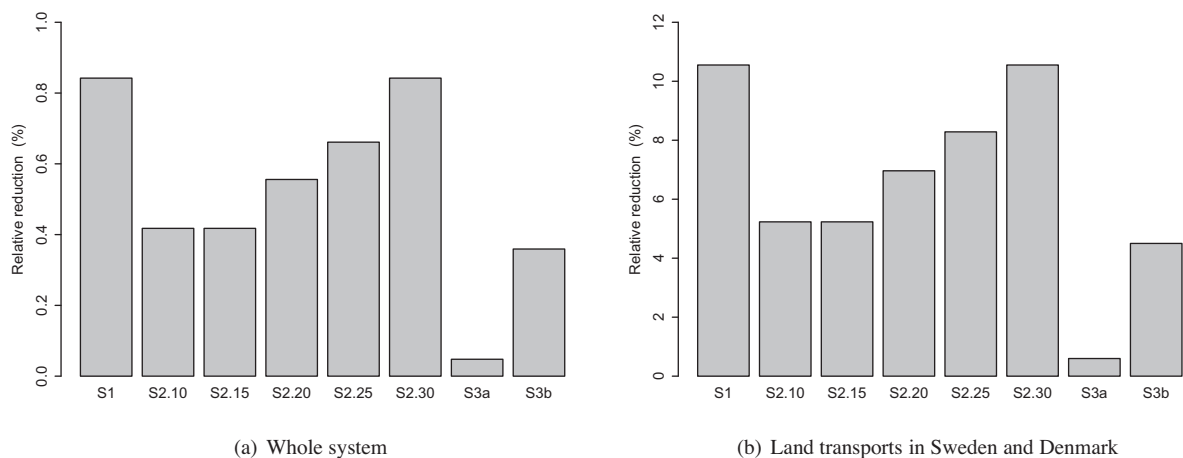


Fig. 5. For each studied measure, the relative reduction (in %) of CO₂ emissions for (a) the whole system, and (b) land transports in Sweden and Denmark.

5. Discussion

A general assumption in TAPAS is that decision makers are local cost minimizers. It is possible to model decision makers in other ways, e.g., by letting them behave according to simple rules that are coded into the agents. However, it is often reasonable to assume that real-world decision makers strive towards behaving optimally. Moreover, it may sometimes be relevant to let other factors than cost, e.g., environmental impact, reliability and punctuality, influence the decision making. However, most types of factors can be represented in the decision making process by different types of costs. For instance, uncertainty regarding reliability and punctuality may be represented as internal costs in organizations, and environmental impact may be represented as governmental taxes and fees.

The cost models that are used in TAPAS (e.g., for production, transportation and ordering) have mainly been chosen to enable real-world resemblance to be obtained in modeled scenarios, and we argue that this is the best way to define cost models in a simulation model like TAPAS. For instance, the model for estimating production costs allows the user to define three types of production strategies; batch production, continuous production, and instant retrieval of products from inventory. Moreover, two different models for setting transport prices are provided, and the user of TAPAS decides on case basis which model is best to use. However, there is room for improvement of the cost and pricing models. For instance, it could be relevant to provide the possibility to model product-dependent fixed costs and preparation times in terminals to be able to represent special requirements (time and other resources) that are needed for loading and unloading of particular product types. Also, it could be relevant to introduce fixed costs in a production line, e.g., to represent wastage when changing product types. However, this would require a more advanced model for determining production costs. Further, it would be relevant to extend and improve the models for calculating transport costs, e.g., since different types of time aspects often have an important influence on the decision making regarding which transport alternatives are most appropriate [36]. A possible extension would be to model punctuality as a cost component to let it have an influence on the decision making. Moreover, train transport costs include many different cost components, which interact in a rather complex way, and a possible extension would be to make use of some of the ideas in the train transport cost model proposed by Troche [43]. Similarly, the ship transport cost model could be further refined, e.g., by also taking into account fuel consumption at berth.

An important advantage of agent models is the modular structure, which enables adjustments of local models. To develop TAPAS to function even better as a general transport chain simulation model, a library of different types of models, strategies, and behaviors would facilitate the simulation of different scenarios. Moreover, the possibility to achieve benefits by advanced cooperation, such as joint planning within the transport chain, is an aspect that would be relevant to investigate further. As an example, the transport providers, as well as the customer could benefit from letting *TB* negotiate directly with the customer regarding the delivery time window when there are cost efficient transport alternatives that have been disregarded due to penalties. To enable cooperation between decision makers to be captured in TAPAS, it would be necessary to extend the proposed interaction protocol. A possible approach would be to model agents as finite state machines, in which different types of behavior are allowed only when an agent is in certain states. This would make it easier to customize the communication between agents for particular scenarios, which would further increase the generalizability of TAPAS.

Another aspect that would be interesting to consider is dynamic agent behavior, which would further improve the possibility to accurately mimic real-world situations. It would be interesting to consider agents that learn from experience how to improve their behavior. For instance, it would be possible for customer agents to improve their ordering strategies by dynamically updating their safety stock levels when it is realized that some product types frequently run out of stock. Moreover, it would be interesting to allow agents to dynamically enter or leave the system, which would make it possible to study how situations evolve over longer periods of time.

6. Conclusion and future work

We have presented a micro-level model for transport chain simulation. By using agent technology, we were able to simulate the decision making activities as well as the interaction between actors, which is difficult, if possible at all, using traditional techniques. By conducting a simulation study in a scenario around the Southern Baltic Sea, we have shown that it is possible to use TAPAS for studying various types of transport policy and infrastructure measures. Moreover, we have shown (1) that it is possible to deal with the complexities of mixing timetabled and demand driven transports in the algorithmic approach (2) that it is possible to simulate the principles of EOQ by letting the customer select the best order quantity among a set of possible quantities, and (3) the interaction framework is appropriate for capturing the ordering process in a multi-agent-based transport chain simulation model for analysis, e.g., of transport-related policy and infrastructure measures.

The studied scenario contains 12 agents; 3 customers, 1 production planner, 5 transport planners, 1 transport chain coordinator, 1 product buyer, and 1 transport buyer. We believe it is possible to scale up the size of scenarios to a few hundred producers and customers without reimplementing TAPAS, however, this need to be confirmed in simulation experiments. The goal is to be able to use TAPAS for studying larger regions containing thousands of producers and customers. A possible approach to improve the scalability would be to enable TAPAS to be executed on multiple processors.

Below we give a few pointers to further development of TAPAS. Extensions of the modeling on the agent level include: (1) integration of sophisticated optimization algorithms in the agents to improve the quality of their decisions, (2) allowing agents to communicate outside the proposed interaction protocol to enable enhanced cooperation between transport chain actors, (3) allowing agents to learn from experience, and (4) allowing agents to dynamically enter and leave the agent system. Examples of interesting extensions to the physical simulator are: (1) including external aspects that affect transportation, e.g., by simulating other traffic on the links, which would enable us to study the effects of link congestion, (2) simulating that real world ordering processing is time consuming (currently order processing is assumed to be instantaneous), and (3) allowing customer orders to be processed in parallel, which would require that aspects related to resource allocation in the ordering process need to be carefully considered.

References

- [1] H. Ahn, P. Kaminsky, Production and distribution policy in a two-stage stochastic push–pull supply chain, *IIE Transactions* 37 (2005) 609–621.
- [2] P. Alstrom, Numerical computation of inventory policies, based on the EOQ/σ_x value for order-point systems, *International Journal of Production Economics* 71 (2001) 235–245.
- [3] S. Åxsäter, *Inventory Control*, second ed., Springer, New York, USA, 2006.
- [4] F.L. Bellifemine, G. Caire, D. Greenwood, *Developing Multi-Agent Systems with JADE* (Wiley Series in Agent Technology), John Wiley & Sons, Chippingham, Wiltshire, England, 2007.
- [5] M. Bergkvist, P. Davidsson, J.A. Persson, L. Ramstedt, A hybrid micro-simulator for determining the effects of governmental control policies on transport chains, in: *Lecture Notes in Computer Science, Multi-Agent and Multi-Agent-Based Simulation*, vol. 3415, Springer, 2005, pp. 236–247.
- [6] J. Boerkamps, A. van Binsbergen, Goodtrip – a new approach for modelling and evaluation of urban goods distribution, in: *Proceedings of City Logistics I*, 12–14 July, Cairns, Australia, pp. 175–186.
- [7] K. Button, *Transport Economics*, second edition, Edward Elgar Publishing Limited, Glos, UK, 1997.
- [8] M. Caramia, F. Guerriero, A heuristic approach to long-haul freight transportation with multiple objective functions, *Omega* 37 (2009) 600–614.
- [9] T.S. Chang, Best routes selection in international intermodal networks, *Computers & Operations Research* 35 (2008) 2877–2891.
- [10] D.C. Chatfield, J.C. Hayya, T.P. Harrison, A multi-formalism architecture for agent-based, order-centric supply chain simulation, *Simulation Modelling Practice & Theory* 15 (2007) 153–175.
- [11] M. Chen, P. Tardieu, The NEAC model, answering questions in a European context, in: *Paper Presented at THINK-UP Seminar 1*, Paris, 2000.
- [12] J.Y.J. Chow, C.H. Yang, A.C. Regan, State-of-the art of freight forecast modeling: lessons learned and the road ahead, *Transportation* 37 (2010) 1011–1030.
- [13] P. Davidsson, J. Holmgren, J.A. Persson, L. Ramstedt, Multi agent based simulation of transport chains, in: *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, International Foundation for Autonomous Agents and Multiagent Systems, May 12–16, Estoril, Portugal, 2008, pp. 1153–1160.
- [14] G. de Jong, M. Ben-Akiva, A micro-simulation model of shipment size and transport chain choice, *Transportation Research Part B* 41 (2007) 950–965.
- [15] G. de Jong, H.F. Gunn, W. Walker, National and international freight transport models: an overview and ideas for further development, *Transport Reviews* 24 (2004) 103–124.
- [16] European Commission, *White Paper – European Transport Policy for 2010: Time to Decide*, Technical Report, Luxembourg, 2001.
- [17] G. Friberg, M. Flack, P. Hill, M. Johansson, I. Vierth, J. McDaniel, T. Lundgren, P. Hesselborn, G. Bångman, *Kilometerskatt för lastbilar – Effekter på näringar och regioner*, Redovisning av ett regeringsuppdrag i samverkan med ITPS, Rapport 2007:2, Swedish Institute for Transport and Communications Analysis (SIKA), 2007.
- [18] H. Friedrich, G. Liedtke, Consideration of logistics for policy analysis with freight transport models, in: *8th Conference on Applied Infrastructure Research*, October 9–10, Berlin, Germany, 2009.
- [19] J. Gjerdrum, N. Shah, L.G. Papageorgiou, A combined optimization and agent-based approach to supply chain modelling and performance assessment, *Production Planning & Control* 12 (2001) 81–88.
- [20] J. Holmgren, An extended eastwest transport corridor (ewtc) scenario, 2011 <[http://www.bth.se/tek/jhm.nsf/attachments/ewtc_pdf/\\$file/ewtc.pdf](http://www.bth.se/tek/jhm.nsf/attachments/ewtc_pdf/$file/ewtc.pdf)>.
- [21] J. Holmgren, P. Davidsson, J.A. Persson, L. Ramstedt, An agent based simulator for production and transportation of products, in: *The 11th World Conference on Transport Research*, June 8–12, Berkeley, USA, 2007.
- [22] J. Hunt, K. Stefan, Tour-based microsimulation of urban commercial movements, *Transportation Research Part B* 41 (2007) 981–1013.
- [23] B. Jansen, P. Swinkels, G.J.A. B.d. Teeuwen, H. Fleuren, Operational planning of a large-scale multi-modal transportation system, *European Journal of Operational Research* 156 (2004) 41–53.
- [24] T. Knabe, M. Schillo, K. Fischer, Improvements to the FIPA contract net protocol for performance increase and cascading applications, in: *International Workshop for Multi-Agent Interoperability at the German Conference on AI (KI-2002)*, 2002.
- [25] V. Komma, P. Jain, N. Mehta, Agent-based simulation of a shop floor controller using hybrid communication protocols, *International Journal of Simulation Modelling* 6 (2007) 206–217.
- [26] G. Liedtke, Principles of micro-behavior commodity transport modeling, *Transportation Research Part E* 45 (2009) 795–809.

- [27] L. Ramstedt, Transport policy analysis using multi-agent-based simulation, Doctoral Dissertation Series No. 2008:09, Blekinge Institute of Technology, Sweden, 2008.
- [28] L. Ramstedt, P. Davidsson, J. Holmgren, J. Persson, On the use of micro-level simulation for estimation of the effects of governmental control policies, in: The 11th World Conference on Transport Research, June 8–12, Berkeley, USA, 2007.
- [29] M.J. Roorda, R. Cavalcante, S. McCabe, H. Kwan, A conceptual framework for agent-based modelling of logistics services, *Transportation Research Part E: Logistics and Transportation Review* 46 (2010) 18–31.
- [30] T.W. Sandholm, V.R. Lesser, Leveled commitments contracts and strategic breach, *Games and Economic Behaviour* (2001) 212–270.
- [31] M. Schillo, C. Kray, K. Fischer, The eager bidder problem: a fundamental problem of DAI and selected solutions, in: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2002), ACM, July 15–19, Bologna, Italy, 2002, pp. 599–606.
- [32] W. Shade, A. Martino, M. Roda, Assessment of transport strategies, in: Proceedings of the 17th International Conference of the System Dynamics Society and the 5th Australian & New Zealand Systems Conference, July 20–23, Wellington, New Zealand, 1999.
- [33] SIKa, Vilken koldioxidskatt krävs för att nå framtida utsläppsmål?, PM 2008:4, Swedish Institute for Transport and Communications Analysis (SIKA), 2008.
- [34] E. Silver, H. Meal, A heuristic for selecting lot size requirements for the case of a deterministic time-varying demand rate and discrete opportunities for replenishment, *Production and Inventory Management* 14 (1973) 64–74.
- [35] R.G. Smith, The contract net protocol: high-level communication and control in a distributed problem solver, *IEEE Transactions on Computers* C-29 (1980) 1104–1113.
- [36] R. Sommar, J. Woxenius, Time perspectives on intermodal transport of consolidated cargo, *European Journal of Transport and Infrastructure Research* 7 (2007) 163–182.
- [37] T. Strader, F. Lin, M. Shaw, Simulation of order fulfillment in divergent assembly supply chains, *Journal of Artificial Societies and Social Simulation* 1 (1998).
- [38] H. Swahn, The Swedish national model systems for goods transport SAMGODS – a brief introductory overview, SAMPLAN-report 2001:1, Swedish Institute for Transport and Communications Analysis (SIKA), 2001.
- [39] J.M. Swaminathan, S.F. Smith, N.M. Sadeh, Modeling supply chain dynamics: a multiagent approach, *Decision Sciences* 29 (1998) 607–632.
- [40] L. Tavasszy, Freight modelling – An overview of international experiences, in: TRB Conference on Freight Demand Modelling: Tools for Public Sector Decision Making, September 25–27, Washington DC, USA, 2006.
- [41] L. Tavasszy, B. Smeenk, C. Ruijgrok, A DSS for modelling logistic chains in freight transport policy analysis, *International Transactions in Operational Research* 5 (1998) 447–459.
- [42] S. Terzi, S. Cavalieri, Simulation in the supply chain context: a survey, *Computers in Industry* 53 (2004) 3–16.
- [43] G. Troche, Activity-based rail freight costing – a model for calculating transport costs in different production systems, Ph.D. thesis, KTH, Royal Institute of Technology, Sweden, 2009.
- [44] D. van der Zee, J. van der Vorst, A modeling framework for supply chain simulation: opportunities for improved decision making, *Decision Sciences* 36 (2005) 65.
- [45] I. Williams, N. Raha, Review of Freight Modelling, Technical Report, Cambridge, UK, 2002. Final Report DfT Integrated Transport and Economic Appraisal.
- [46] R. Wilson, A scientific routine for stock control, *Harvard Business Review* 13 (1934) 116–129.
- [47] M.J. Wooldridge, N.R. Jennings, Intelligent agents: theory and practice, *The Knowledge Engineering Review* 10 (1995) 115–152.
- [48] W. Xuehui, Z. Lei, X. Nong, T. Yuhua, Time management in parallel discrete event simulation, in: International Forum on Information Technology and Applications (IFITA 2009), IEEE, May 15–17, Chengdu, China, 2009, pp. 209–212.
- [49] H. Zhou, W. Benton Jr., Supply chain practice and information sharing, *Journal of Operations Management* 25 (2007) 1348–1365.
- [50] A. Ziliaskopoulos, W. Wardell, An intermodal optimum path algorithm for multimodal networks with dynamic arc travel times and switching delays, *European Journal of Operational Research* 125 (2000) 486–502.