Synchromodal Transport Planning at a Logistics Service Provider

Martijn R.K. Mes and Maria-Eugenia Iacob

Abstract In this chapter, we consider synchromodal planning of transport orders with the objective to minimize costs, delays, and CO_2 emissions. Synchromodal planning is a form of multimodal planning in which the best possible combination of transport modes is selected for every transport order. The underlying problem is known as the multi-objective k-shortest path problem, in which we search for the k-shortest paths through a multimodal network, taking into account time-windows of orders, schedules for trains and barges, and closing times of hubs. We present a synchromodal planning algorithm that is implemented at a 4PL service provider located in the Netherlands. We illustrate our approach using simulation with order and network data from this logistics service provider. On the corridor from the Netherlands to Italy, an average cost reduction of 10.1 % and a CO_2 reduction of 14.2 % can be achieved with synchromodal planning.

Keywords Synchromodal transport · Intermodal transport · Transportation planning · Shortest path problem · Decision support systems

1 Introduction

Growing freight flows, increasing road congestion, and increasing pressure to lower emissions all stimulate the use of intermodal transport. Commonly, in intermodal transport different modes of transport are decided upfront for certain shipment types, such as shipments on a particular corridor. Since the actual transport orders and network characteristics may vary (e.g., rush orders and low water levels), a safe choice is often to rely on road transport. A recent development to cope with the inflexibility of pre-determined modes is the use of synchromodal transport. Here,

M.R.K. Mes (✉) · M.-E. Iacob
Department of Industrial Engineering and Business Information Systems,
University of Twente, Enschede, The Netherlands
e-mail: m.r.k.mes@utwente.nl

H. Zijm et al. (eds.), *Logistics and Supply Chain Innovation*,
Lecture Notes in Logistics, DOI 10.1007/978-3-319-22288-2_2

the best possible combination of transport modes is selected dynamically for each incoming order, based on aspects like costs, duration, reliability, and sustainability. A shipper agrees with a logistics service provider (LSP) on the delivery of products at specified costs, duration, and sustainability but gives the LSP the freedom to decide on how to organize the transportation according to these specifications. The LSP is able to deploy different transport modes flexibly depending on real-time information, traffic conditions, and resource availability. Synchromodal planning offers the opportunity to improve transportation services in terms of sustainability, costs, and quality.

In this study, we investigate the applicability of synchromodal transport planning at a Dutch 4PL service provider active in the European transport market. An exploration of the concept of synchromodal planning reveals its high complexity compared to truck planning. The option to select from more alternatives also increases the demand for computer aided planning. Furthermore, a potential increase of stakeholders, like multiple carriers and terminal operators, leads to higher planning complexity and also requires more coordination tasks for the LSP.

We propose a planning algorithm with the objective to minimize costs and CO_2 emissions, while maintaining delivery reliability. Our goal is to provide the human planner with a list of the top k multimodal routes. The complexity of our algorithm depends on the number of legs available for transport, where a leg is defined as a combination of a direct connection between two hubs (points where a modal switch can take place), a carrier operating on this connection, and—possibly—a pre-defined departure and arrival time.

The goal of this chapter is twofold: First, we propose a synchromodal planning algorithm that is easy to work with, easy to implement, and able to cope with many practical challenges. Second, we illustrate and test our approach using a real case study, thereby showing the potential benefits of synchromodal planning.

The remainder of this chapter is structured as follows. In Sect. 2 we provide a literature review on planning methods for synchromodal transport. After introducing the problem in Sect. 3, the synchromodal planning approach is proposed in Sect. 4. In Sect. 5, we shortly describe the implemented system and how it is used in practice. Our analysis and the results of the case study can be found in Sect. 6. We end with conclusions and recommendations (Sect. 7).

2 Literature

Synchromodal transport is closely related to intermodal transport. Literature reviews on intermodal transport are given by Macharis and Bontekoning (2004), Caris et al. (2008), and SteadieSeifi et al. (2014). They all conclude that the majority of research on intermodal transport focuses on strategic decisions, such as network design and location of terminals. Operational aspects, like mode choice and routing decisions for incoming shipments, have received limited attention

(Caris et al. 2008). We mention some related studies on operational aspects in intermodal transport.

Boardman et al. (1997) describe a decision support system to determine the best combination of transport modes to use for intermodal shipments. Their system evaluates multiple routes based on costs, service level, and type of commodity being shipped. Ziliaskopoulos and Wardell (2000) consider a similar situation, but explicitly take into account transfer times and time-dependent arc travel times; the latter being derived from transit time tables. Also in the area of public transport, some related work can be found. For example, Horn (2004) considers a traveller information system that provides a sequence of journey-legs (combination of walking, taxi, bus, and train) that meet the traveller's requirements against lowest possible costs. The proposed planning procedure first generates routes consisting of only a single leg, and gradually increases the number of legs in a route while simultaneously establishing upper bounds. Horn (2004) states that this procedure generally leads to an optimal schedule quickly, because good routes tend to contain only few legs. We also utilize this idea, since in intermodal transport, each transhipment results in additional handling costs.

The planning problem we consider is related to the *k*-shortest paths problem (Eppstein 1998). Our goal is to provide the human planner with a list of the top *k* multimodal routes taking into account costs, CO_2 emissions, and delivery reliability. Many of the *k*-shortest path algorithms, such as those proposed by Eppstein (1998) and Yen (1971), provide good results, but their running time complexity is quite large, which is inconvenient in practical situations when the available time for calculating a new set of options might be very short. Vanhove and Fack (2012) propose a heuristic that is capable of doing this, although there is no absolute guarantee that the *k* shortest paths are always found.

Shortest path problems typically have one objective, namely minimizing the travel distance. Typically, for barges and trains, the costs and CO_2 emission per kilometre would be lower compared to trucks; however, the total transport time is usually larger. Therefore, our problem can be best characterized as a multi-objective *k*-shortest paths problem, because we have to balance costs, emissions, and delivery punctuality. A few existing approaches are relevant in this context. Martins (1984) presents two algorithms, tested on a shortest path problem, in which multiple objectives can be included. Climaco et al. (1982) presents an algorithm that returns the set of Pareto optimal paths. An algorithm to solve bi-criterion shortest path problems is developed by Mote et al. (1991); this algorithm also returns the set of Pareto optimal paths. Drawback of this method is that only two parameters can be taken into account for each leg. Mote et al. (1991) have tested their method against the *k*-shortest path approach of Yen (1971); they conclude that their parametric approach is significantly faster.

Besides the requirement to provide the planner multiple options and select these options based on multiple criteria, additional complexity is introduced in our problem due to (i) the decision which carrier to use on which leg at what time, (ii) soft time windows, (iii) pre-defined timetables for rail and barge transport causing variable waiting times, (iv) contractual agreements, (v) complex costs functions depending on time, weight and volume, and (vi) the possibility of freight consolidation. The contribution of this chapter is that we provide a multi-objective *k*-shortest path algorithm that is able, in combination with the human planner, to handle the afore-mentioned practical complexities. In addition, we describe the planning system that has been implemented at the LSP, and provide numerical results quantifying the benefits of synchromodal planning at this LSP.

3 Problem Description

Our problem concerns an LSP that provides logistics services (i.e., transportation and warehousing) to its customers. These customers (i.e., the shippers) have freight that needs to be transported from a pickup location to a delivery location. The pickup is only possible after an earliest pickup time, and delivery after a latest delivery time is penalized. A lane is defined as the combination of a pickup and delivery point. The LSP has contracts with multiple carriers that have one or more modalities available to transport the goods.

The network where carriers operate consists of hubs and legs. Hubs are points where containers can be transferred from one modality to another. Legs are route segments between two hubs. Since multiple carriers might operate on the same leg, we introduce the term carrier-leg. This can be defined as the leg a carrier is assigned to with the related values of the criteria (e.g., costs). We extend this concept by adding another level by connecting the resource (e.g., the train leaving at 10 a.m.) to it: resource-carrier-leg. We make a distinction between main legs and free legs. Main legs have a fixed schedule (in location and time), whereas free legs offer more flexibility. A route consists of a sequence of legs connecting the pick-up and delivery point of a transport order. In case of 'traditional' road transport, we only have a free leg connecting the origin with the destination (left part of Fig. 1),

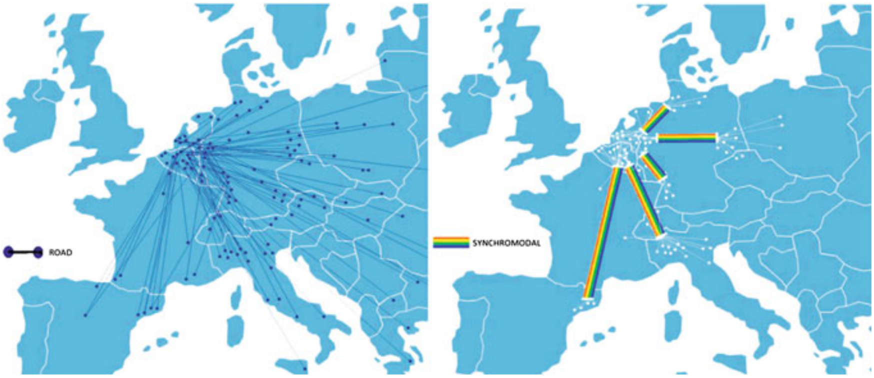

Fig. 1 Example of road transport versus intermodal transport

whereas in intermodal transport, the route typically consists of a few main legs connected by free legs (right part of Fig. 1). The challenge is to find, for each incoming order, *k* routes that score well on the criteria costs, duration, and emission.

To simplify the presentation of the planning algorithm, we make some assumptions. Later on, in Sect. 5, we describe how to generalize these assumptions in practice. First, we assume that all costs, times, and emissions are known and deterministic. This holds for the legs, but also for hubs that have closing times and transhipment costs. Second, we ignore contractual restrictions, such as volume agreements and preferred carriers. Finally, orders are planned one by one, without considering consolidation opportunities.

Our proposed algorithm will be used within a control tower. A control tower is software that helps a planner to direct the transport through the network. It takes restrictions into account and tries to find the best way (based on user preferences) to schedule all transports. Next to this, the control tower provides all users with real-time data about all ongoing transport orders.

4 Approach

In our intended implementation environment, we face several challenges. First, the optimization criteria and constraints are ‘negotiable’, which means that the exact objective coefficients (weights for various optimization criteria) are not given up front, and constraints on duration (delivery windows) and contracts (volume agreements) are soft (flexible). Second, the human planner does not want a single ‘optimal’ recommendation for multi-modal transport, but wants to choose from a top *k* of potentially good options, such that the planner can balance various optimization criteria as well as constraint violations (through sorting and filtering). Given these characteristics, we decide to use a sequential heuristic approach.

Our approach consists of 2 stages: (i) offline steps in which no order-specific characteristics are taken into account, and (ii) online steps in which order-specific characteristics are taken into account, to make sure that the generated options are suitable for a given order at a given time.

In the offline steps, also called pre-processing, we attempt to reduce the network for each lane (combination of origin hub and destination hub), which will speed-up the process of determining the top *k* options when an order arrives. Pre-processing starts from a full network including all legs and hubs, an origin hub and a destination hub. By removing hubs (and their associated legs) from this network, which are geographically too far from the route between the origin and destination hub, a reduced network for a specific lane can be created. We use a parameter α to indicate a threshold on the maximum increase in distance when inserting a hub in a route from the origin to the destination hub (i.e., with $\alpha = 1.5$, we allow a 50 % increase in total distance when adding a hub). The resulting reduced network is a subset of the complete transport network, and legs that are likely not to be used for this order are not part of this subnetwork. The trigger for executing these offline steps does not

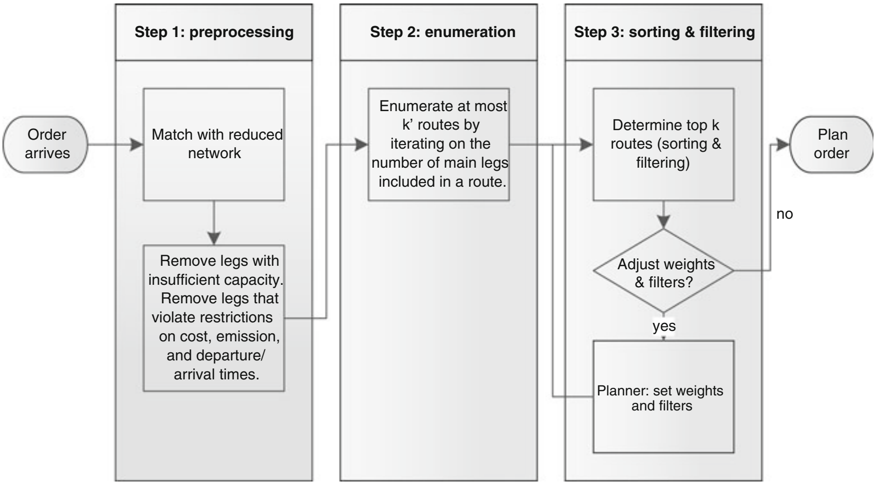
Step 1: preprocessing
Step 2: enumeration
Step 3: sorting & filtering
Order arrives
Match with reduced network
Remove legs with insufficient capacity. Remove legs that violate restrictions on cost, emission, and departure/ arrival times.
Enumerate at most k' routes by iterating on the number of main legs included in a route.
Determine top k routes (sorting & filtering)
Adjust weights & filters?
no
yes
Planner: set weights and filters
Plan order

Fig. 2 General overview of the approach

have to be an incoming order, but may also result from a change of the network. The smaller the reduced network is, the faster the online steps can be carried out.

The online steps are triggered when an order arrives. We distinguish between three steps, as shown in Fig. 2. In the remainder of this section, we explain the algorithm in more detail.

In the first step, the incoming order will be matched with a certain lane (or a few lanes) so that a reduced network is selected (the selected lane has a pickup hub closest to the pickup location of the order and a destination hub closest to the delivery location of the order). This network can be reduced even further at this step, because order characteristics are known. Legs with insufficient capacity are removed. For all remaining main legs, resource-carrier-legs will be generated (see the definition in Sect. 3) for each possible departure time within a given planning horizon (e.g., the period between the earliest pickup time of the order and a few days after latest delivery time). In the remainder, we simply refer to these time-resource-carrier-legs as *main legs*.

The next step is to enumerate all possible combinations of main legs (which result in routes after adding free legs where necessary) to determine, for given weights, a relative large number k' of 'shortest' routes, with $k' \gg k$. In the third step, the human planner can apply filters (filtering out less desirable results taking into account criteria not explicitly included in the planning algorithm) and change the weights attached to the three criteria. After changing the filters and weights, the results are sorted and at most k routes are shown.

The first and third step can be performed very fast (in milliseconds). Note that even for a large value of k', the third step only requires (i) selecting the k-shortest paths and (ii) sorting these k paths, i.e., making sure that the remaining paths are worse than the *k*th path. However, the enumeration step (second step) might cause computational complexities.

In the enumeration step, we iterate on the number m of main legs included in a route. We do this because (i) good routes are likely to have only a few main legs and (ii) during iteration we can check whether it is efficient to extend existing routes with additional main legs. An enumerated route is characterized by its sequence of main legs. Free legs are added to these routes to make sure they are connected (connected path from pickup to delivery location through all hubs). In case there are multiple free legs available, we choose the one with lowest costs (alternatively, the cheapest one based on default settings of the weights).

In iteration m, we take all routes from iteration $m - 1$ (routes with $m - 1$ main legs), and for each of these routes we remove the last free leg connecting to the delivery location of the order. A route up to the last hub (i.e., a route excluding the last free leg connecting to the delivery location) is called a partial route. In an iteration, we loop over all partial routes and loop over all possible main legs to be added at the end of these partial routes. Possible main legs to add are those that have (i) a destination hub not yet present in the route and (ii) a closing time later than the earliest arrival time at the origin hub. In case there are multiple resource-carrier-legs of the same carrier-leg (different departure times within our planning horizon), we only consider the first suitable departure time. For each of the extended partial routes, we add free legs where necessary (to create a connected path from pickup to delivery location).

At the end of each iteration, we check which partial routes are eligible for consideration in the next iteration. For this, we introduce two thresholds: β and γ. These thresholds represent the maximum allowable increase in costs (β) and delay (γ) of the partial route compared to respectively the cheapest route and the route with the least delay found so far. After updating the lowest costs or lowest delay, we will be more critical in adding new routes. However, we might also come back to routes generated earlier that now might become infeasible regarding the maximum deviation to the lowest costs (β) and to the lowest delay (γ). In addition to these two thresholds that limit the number of routes enumerated, enumeration is also limited to k' routes at most.

Obviously the number of routes explodes due to the number of main legs. To illustrate this, suppose we have M main legs in a reduced network; then we already have *M!* possible routes containing all of the M legs (we have a full connected graph since trucks can drive between all hubs). The total number of routes R of all possible lengths m is be given by:

$$R = \sum_{m=0}^{M} \frac{M!}{(M-m)!}$$

For a network with $M = 35$ carrier-legs, a typical amount faced by the LSP under study, we have 2.8×10^{40} different routes (note that our algorithm uses the resource-carrier-legs as input, which automatically results in even more routes).

Obviously, the majority of these routes will contain many main legs, whereas in practice one would expect the better routes to contain only a few main legs (less transhipment costs). For the example with 35 legs, we have approximately 1.2 million routes containing at most 4 main legs. In addition, the number of routes is reduced using the thresholds β and γ.

The values of the thresholds have a large impact on the output of the algorithm. Setting appropriate vales for them is therefore important. We use three tuneable parameters: α (used in the offline steps), β (threshold on the maximum costs of a partial route), and γ (threshold on the maximum delay of a partial route). In the remainder of this section, we discuss how to find appropriate values.

It is difficult for the human planner to determine good values for the thresholds as it is hard to estimate the impact of certain values, e.g., on the number of routes R that will be generated. Setting the thresholds too high results in long computation times, while setting the thresholds too tight results in limited options to choose from. Also note that the human planner needs enough options because (i) he has to balance the three criteria (changing the weights in step 3 of our planning approach) and (ii) has to apply other criteria not explicitly included in the enumeration step (using the filters to, e.g., to filter out results that violate contractual agreements). Obviously, when no additional criteria play a role, β can be set to 1, i.e., only accepting the 'shortest' route for the given weights.

With respect to α, it is useful to have an α that is lane-specific; lanes with an origin and/or destination in densely populated areas will have a lower α than when they connect sparsely populated areas. This way we can make sure that for each lane a comparable number of hubs remains in the reduced network, which is likely to result in a comparable number of routes to enumerate. We can assist the human planner by showing a map that displays the result for a given α. In addition, we can use a feedback method: when the number of enumerated routes exceeds k' for a certain lane, we advise the planner to reduce α for this lane. A possible extension is to use an automated approach, where all possible realizations of the number of enumerated routes and used threshold per lane/corridor are stored, and used to assist the human planner in the best choice of parameter settings. Whenever a planner (or system administrator) changes the values of a threshold, the user is provided with feedback on the expected number of routes that will be enumerated. If this exceeds the limit k', visual warnings can be given. Moreover, the system may learn to propose the 'right' set of k routes by memorizing the routes most often selected by the human planner. If the human planner always selects a route consisting of hubs close to the direct route (hence which would have been selected with a low value of α), the value of α could be decreased. This will make future calculations faster and more efficient.

The same line of reasoning for setting α can also be followed for setting β and γ. The right choice for these parameters depends on k'. Ideally, β and γ are set in such a way that a number close to k' routes are generated. To give an indication of reasonable parameter settings, we used the following values in our case study: a value of $k' = 100{,}000$ (a lower value $k' = 10{,}000$ did not provide a sufficiently large number of results to handle all additional constraints taken care of in the filtering step), $\alpha = 2$ (accepting hubs that result in routes twice as long as a direct connection), $\beta = 2$ (accepting partial routes that are twice as expensive as the cheapest route found so far), and $\gamma = 2$ (accepting routes that have 1 additional day of lateness compared to the route with the least amount of lateness).

5 Relaxation of Assumptions

In Sect. 3, we made several assumptions for our algorithm. Specifically, we assumed that (i) input data is known and deterministic, (ii) no contractual restrictions have to be taken into account within the algorithm, and (iii) only one job at a time is considered. In this section, we describe how to relax these restrictions in practice.

The proposed algorithm treats the current status of the network as static and deterministic input. However, execution of plans might work out differently than planned. We might face unexpected delays, e.g., caused by traffic jams, missed connections, and breakdowns. In these situations, we have to perform re-planning. The system can be used by modifying the changed network status (e.g., removing legs or changing the duration of legs) and planning the order from its current location onwards. The main challenge here is to adjust, preferably automatically, the input data used by the algorithm.

We previously mentioned that the existence of contracts and governmental regulations is not considered within the algorithm. However, contractual restrictions can be taken into account in the sorting and filtering step. The benefit of this approach is that the algorithm itself (i.e., the enumeration step) remains fully independent of contractual changes. Restrictions like required legs and carriers can be incorporated both in the pre-processing stage (by changing the input data) and the sorting and filtering phase. Regarding the latter, if a leg or carrier is required, we can simply filter out all routes that violate this requirement. If we have volume contracts (e.g., on a certain corridor, 50 % of the loads should be rewarded to carrier A and 50 % to carrier B), we have another performance criterion that states for each route the deviation from contractual agreements. We can then filter out large deviations, or we can weigh this additional criterion against the others.

Finally, our assumption to schedule one job at a time makes it hard to consolidate jobs. Even though we typically deal with containers in intermodal transport, consolidation might be possible. Again, by changing the input data, we can support consolidation decisions with the implemented planning system. In the time between planning and execution of an order, we can create ‘copies’ of the legs planned for this order. These copies have reduced capacity and possibly a reduced price. In the planning of a subsequent order, we also consider these copies, thus enabling consolidated shipments on certain legs.

6 Implementation

In this section, we shortly describe the Synchromodal Control Tower (SCT) system and its relationship with other surrounding systems from the LSP and with external actors. The structure of the business network served by the SCT is depicted in Fig. 3. The system needs inputs from the shippers and the warehouses where the cargo will be loaded or dropped. The planning is communicated with the carriers handling the cargo on the different legs along the chosen lanes. The system is also connected with the customs for the automated handling of paper work.

The functional components of the SCT's application architecture are shown in Fig. 4. In the old situation, the LSP's architecture consisted only of the software components shown in Fig. 4 in blue or green. More precisely, the LSP had an enterprise resource planning system in place supporting all operational tasks (e.g., the static planning of shipments and invoicing), a Geographical Information System for the mapping and routing of shipments, and a Track and Trace module for the monitoring of ongoing transports. In the new situation, a synchromodal control tower application (shown in red) has been added. Due to this major change and the need to establish connections between the new tower and the existing systems, some of the existing application components (those coloured green) required some modifications in order to make the exchange information with the SCT possible. Furthermore, the architecture also shows the external parties that have access, or provide information to the new SCT.

The SCT consists of several components, the most critical one being "Top k lanes calculation". The main task of this component is to carry out the calculation of the *k*-shortest routes, based on the algorithm explained in Sect. 4, and using a

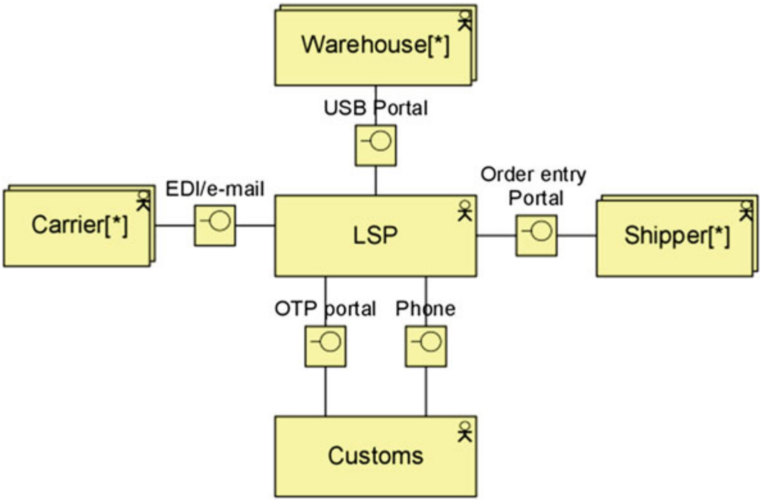

Fig. 3 Actor model

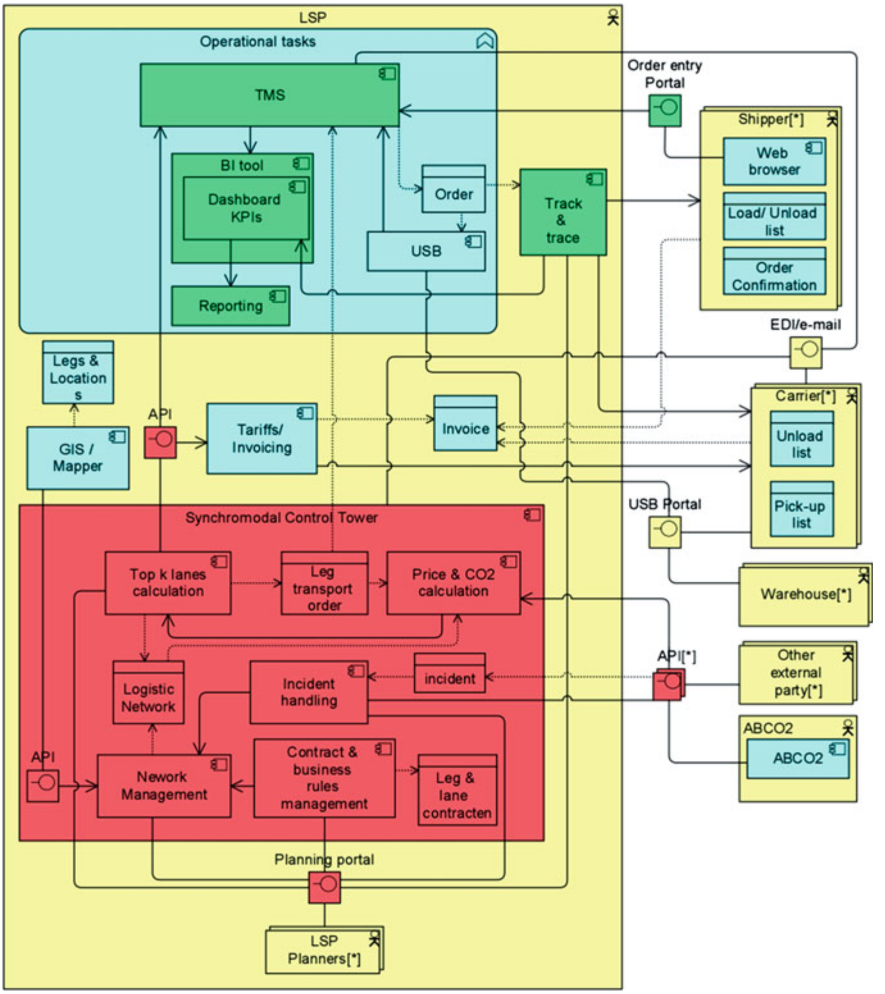

Fig. 4 Application architecture

network of legs and hubs as input (the management of which is done by the "Network management" component).

Another important function of the SCT concerns the price and CO_2 emissions calculation for the chosen lane, which is dealt with through the "Price & CO_2 calculation" component. For the CO_2 emission calculation, we use an external, cloud based, carbon management system ($ABCO_2$) that has been designed on the basis of the reference architecture proposed by Iacob et al. (2013).

The implementation of the SCT has also lead to better "Track and Trace" and business performance monitoring (i.e., the "BI tool") functions. Orders can be traced now (nearly real time) up to leg-level, and the dashboard of the BI tool has been enriched with new KPI's to monitor the carrier's performance per leg.

A number of new interfaces had to be built to ensure the exchange of information with internal and external systems and users. The most important one enables the human planner to interact with the planning algorithm, and has functions such as setting values for parameters and running the algorithm, filtering of orders and lanes, management of contracts, management of the logistic network and of incidents, etc.

7 Validation

We illustrate and test our approach using an order set from the LSP. The LSP currently outsources its transport per lane (combination of pickup and delivery point) to a preferred carrier (typically truck transport). In principle, all transport on this lane will be outsourced to this carrier. In the new synchromodal way of working, this LSP will decide upon the carriers in a dynamic way. Obviously, this makes the planning process more complex, but they aim to realize significant efficiency gains with respect to transport costs and CO_2 reduction. The synchromodal planning system described before is consistent with their wishes and can therefore be used. To determine the impact of the planning approach, a simulation study will be carried out.

The goal of the simulation study is to provide insight into the working of the synchromodal planning approach. With the simulation study, the approach can be clearly explained to the planners of the LSP, which will convince them of its power and will help in the process of accepting this new way of working. The main indicators are the costs, CO_2 emission, and service level. Next to this, the simulation study will also be used to validate the planning algorithm.

For our experiment, we focus on the corridor between the Netherlands and the middle of Italy, including Germany, Belgium, Switzerland and Austria. We use orders on this corridor during one quarter of 2013, originating in The Netherlands (1728 in total). We use the historical data set to determine the following characteristics. For the orders, we use the pickup and delivery location, earliest pickup time, latest delivery time, and volume. For the network information, we use 110 available barge and rail connections (legs) on this corridor. For each leg, we define the carrier, modality, distance, CO_2 emission, schedule of departure and closing times, and duration. For each hub, we use the coordinates, opening times, possible modality switches, and handling and transfer times. As costs components we use a price per kilometre for each carrier-leg, and for the handling- and transhipment costs per hub.

The results showed that for 39 % of the orders it is beneficial to make a modal shift (for at least part of the route) from road to rail (30 %), barge (5 %) or a combination of both (4 %). Mainly orders with destination in Italy, Switzerland and Austria showed to be suitable for synchromodal planning because of their relative long transport distance and good connections over rail and water (we observe a minimum distance of approximately 400 km before intermodal transport is preferred over direct truck transport). On average, a cost reduction of 10.1 % can be achieved with synchromodal planning, while CO_2 emissions reduce with 14.2 %, which comes down to a yearly reduction of 712 tons of CO_2 on this single corridor. Additional savings can be achieved by extending the deadlines with 1 day (1.5 % cost reduction) and by limiting the FTL size of orders to the capacity of 40 ft containers (4.0 % cost reduction).

The results so far are based on the assumption that the LSP does not consolidate shipments, which in practice does not hold. For a fair comparison, we extend the synchromodal planning algorithm with a consolidation option (see Sect. 5). In this case, the cost reduction of synchromodal transport with consolidation compared to truck transport with consolidation is 3.9 %. This value is lower than the original reduction of 10.1 %, because consolidation opportunities arise when there is sufficient slack in the delivery windows. Since intermodal transport typically takes longer than truck transport on this corridor, the cost reduction of synchromodal transport gets smaller.

8 Conclusions

We developed an algorithm to support synchromodal transport planning. The underlying problem is known as the multi-objective k-shortest path problem, in which we search for the k-shortest paths through a multimodal network taking into account time windows of orders, schedules for trains and barges, and closing times of hubs. We provide the human planner with a list of the top k multimodal routes, with the objective to minimize costs, delays, and CO_2 emission.

Our approach consists of an offline phase and an online phase. In the offline phase, also called pre-processing, we attempt to reduce the network for each lane (combination of origin hub and destination hub). The online phase is triggered when an order arrives. We distinguish between three steps. In the first step, the incoming order will be matched with a certain lane so that a reduced network is selected. The next step is to enumerate all possible routes from origin to destination within the reduced network. In this step, we iterate on the number of main legs included in a route, since routes with a limited number of transfers tend to be better. In the third step, the human planner can apply filters, change the objective weights, and make the final choice.

We illustrate our approach using simulation with order and network data from a Dutch 4PL service provider. On the corridor from the Netherlands to Italy, an average cost reduction of 10.1 % and a reduction in CO_2 emission of 14.2 % can be achieved with synchromodal planning.

The proposed synchromodal algorithm has been implemented at the LSP within their supply chain control tower. This system contains all required information on the LSP's main transport routes (time tables, costs, duration, emission, etc.), provides real-time decision support on the best combination of transport modes, and monitors the actual execution of transport. The synchromodal control tower was put into use in 2013, and already during the second half of 2013 a modal shift of 1500 TEU with a CO_2 reduction of 3230 ton has been reported.

References

Boardman BS, Malstrom EM, Butler DP, Cole MH (1997) Computer assisted routing of intermodal shipments. Comput Ind Eng 33(1):311–314

Caris A, Macharis C, Janssens GK (2008) Planning problems in Intermodal freight transport: accomplishments and prospects. Transp Planning Technol 31(3):277–302

Climaco J, Martins E (1982) A bicriterion shortest path algorithm. Eur J Oper Res 11(4):399–404

Eppstein D (1998) Finding the k shortest paths. SIAM J Comput 28(2):652–673

Horn ME (2004) Procedures for planning multi-leg journeys with fixed-route and demand responsive passenger transport services. Transp Res Part C Emerg Technol 12(1):33–55

Iacob ME, van Sinderen MJ, Steenwijk M, Verkroost P (2013) Towards a reference architecture for fuel-based carbon management systems in the logistics industry. Inf Syst Front 15(5): 725–745

Macharis C, Bontekoning YM (2004) Opportunities for OR in intermodal freight transport research: a review. Eur J Oper Res 153(2):400–416

Martins E (1984) On a multicriteria shortest path problem. Eur J Oper Res 16(2):236–245

Mote J, Murthy I, Olson D (1991) A parametric approach to solving bicriterion shortest path problems. Eur J Oper Res 53(1):81–92

SteadieSeifi M, Dellaert NP, Nuijten W, Van Woensel T, Raoufi R (2014) Multimodal freight transportation planning: a literature review. Eur J Oper Res 233(1):1–15

Vanhove S, Fack V (2012) An effective heuristic for computing many shortest path alternatives in road networks. Int J Geogr Inf Sci 26(6):1031–1050

Yen J (1971) Finding the K shortest loopless paths in a network. Manage Sci 17(11):712–716

Ziliaskopoulos A, Wardell W (2000) An intermodal optimum path algorithm for multimodal networks with dynamic arc travel times and switching delays. Eur J Oper Res 125(3):486–502