Transportation Research Part C 17 (2009) 149–162

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Integration of vehicle routing and resource allocation in a dynamic logistics network

Thomas Huth *, Dirk C. Mattfeld

Decision Support Group, Carl-Friedrich Gauß Department, Braunschweig Institute of Technology, Abt-Jerusalem-Straße 4, D-38106 Braunschweig, Germany

ARTICLE INFO

Article history:
Received 15 July 2007
Received in revised form 14 July 2008
Accepted 15 July 2008

Keywords:
General pickup and delivery problem
Dynamic resource allocation
Swap container
Model integration
Multi-stage transportation problem

ABSTRACT

A shipper plans daily hub-to-hub transports within a hub and spoke network. Since a limited number of swap containers is available for transportation, two problems arise. 1. Swap containers have to be routed as pickup and delivery requests in multi-hub tours. 2. Day-by-day routing may lead to an imbalance of swap containers requiring a dynamic allocation. Neglecting interdependencies between vehicle routing and resource allocation seems inferior. An integration of the two problems overcomes this deficiency. We formulate mathematical models and propose integration approaches. The advantages of these approaches are discussed based on a computational study.

1. Introduction

We refer to a real-world routing and allocation problem adapted to the needs of an academic treatment denoted as *swap container problem* (SCP) in the following (Huth and Mattfeld, 2007a,b). The SCP considers the integration of day-by-day routing decisions in a hub-to-hub transportation network and the allocation of empty swap containers to hubs over time.

Geoffrion proposes a generic concept of model integration (Geoffrion, 1989, 1999). The *deep integration* approach merges two models into a new embracing one. A weaker connection of models is implemented by the *functional integration* passing information from one model to the other. We apply both concepts to the SCP.

The outline of this paper is as follows. In Section 2, we describe the SCP and provide mathematical formulations for a separate solution strategy. Integration approaches for the SCP are proposed and discussed in Section 3. In Section 4, a large neighborhood search algorithm is introduced and applied to the integration approaches proposed. Computational results for several test instances are presented in Section 5. 6 concludes the paper.

2. Problem description and modeling

2.1. An adapted real-world problem

Today, large parcel service providers (shipper) operate in hub and spoke networks (Grünert and Sebastian, 2000). Customers are located in catchment areas of hubs serviced by spokes. In this paper we consider daily hub-to-hub transports occurring between all hubs. For transportation swap containers are used. These are self-contained transportation units with a

* Corresponding author. Tel.: +49 531 391 3215; fax: +49 531 391 8144.
E-mail address: t.huth@tu-bs.de (T. Huth).

0968-090X/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.trc.2008.07.004

standardized size of 24.5 foot length, 8.2 foot width and a maximal total weight of 16 tons. Swap containers can be deposited easily by pulling out the foldable put-down feet. One loaded swap container represents one *transportation request*. A truck trailer can transport at most two swap containers.

Between every pair of hubs, typically one or more swap containers are shipped by one or more trucks in a period. Transports are performed by third party carriers paid on the basis of transportation distances only. One task of the shipper is the construction of routes between hubs so that transportation requests can be reasonably combined. Whenever only one swap container is transported by a truck, a slight bypass via another hub and the entrainment of a second swap container from this hub may result in a benefit compared to two direct shipments. This feature is referred to as *detouring and entrainment* in the following.

Although a sufficient number of swap containers is available in the network, another task is to allocate swap containers at the hubs over time. Swap containers are allocated in accordance to the known demand of the forthcoming periods. To satisfy the demand at a hub for a point in time, the currently available swap containers minus outgoing requests (pickups) plus incoming requests (deliveries) are calculated. A demand of a hub can typically be satisfied from different supply hubs, thus a reasonable supply hub is to be selected. In contrast to transportation requests, we refer to the resulting supply–demand pairs as *allocation requests*.

Traditionally, tasks are performed sequentially in the course of time as shown in Fig. 1. A shipper's day begins with the loading of swap containers in the evening. At night the transportation requests are routed by external carriers and are unloaded at the delivery hubs in the morning. Whenever there is a lack of empty swap containers for the next period, the demand exceeds the available amount, deadheads during the day are performed to obtain a balance in the network. These costly deadheads may be saved by integrating the allocation requests in the preceding routing phase, respectively. In this paper we propose the transport of empty swap containers as a by-product of routing by means of detouring and entrainment.

2.2. Model building

Traditionally, both problems during the shipper's day are modeled and performed separately in an alternating sequence of transports and deadheads. The routing problem can be modeled as a generalization of the *Pickup and Delivery Problem* (PDP) whereas the allocation problem can be modeled as a multi-stage *Transportation Problem*. A large body of literature is available for each of these two problems.

A recent survey summarizes the state of the art on pickup and delivery problems (Parragh et al., 2006,). In the literature on container allocation, loaded and empty flows of resources are considered independently (Cheung and Chen, 1998, 2002, 1993, 2005, 2008). As suggested in (Dejax and Crainic, 1987), the independent consideration neglects possible synergies arising from an integrated view on these problems. The survey paper of (Dejax and Crainic, 1987) provides a comprehensive review on empty flows in freight transportation, a taxonomy, modeling and solution techniques.

2.2.1. Pickup and delivery model

This section provides the mathematical formulation of the General Pickup and Delivery Problem (GPDP), inspired by (Savelsbergh and Sol, 1995). Despite some enhanced characteristics of our problem, we do not have to adapt the model substantially. The adapted set construction improves the performance and readability of the model. Modifications in the data structure cover all requirements.

The GPDP is defined on a graph G containing nodes N and edges E. All hubs are in N. A transportation request $a \in \mathrm{TR}$ consists of a pickup node r_a^+ and a delivery node r_a^-. The set of pickup and delivery locations are determined by the pickup and delivery nodes of all requests $R^+ := \bigcup_{a\in \mathrm{TR}}\{r_a^+\}$ and $R^- := \bigcup_{a\in \mathrm{TR}}\{r_a^-\}$. $R = R^+ \cup R^-$ and $R \subseteq N$.

The trucks $k \in K$ have an origin depot and a destination depot. Since each node can be considered as a depot, the problem turns into a general pickup and delivery problem. The origin depots of trucks are located in the origin node k^+ and the destination depots are located in the destination node k^-. This can be described as follows $D^+ := \{k^+ | k \in K\}$ and $D^- := \{k^- | k \in K\}$. $D = D^+ \cup D^-$ and $D \subseteq N$.

A multi-digraph representation promises a more efficient formulation of the model by arc set construction. Therefore, we introduce three more sets. Let A^k be the set of feasible arcs for the route of truck k. For each node i the set $O(i, k)$ represents the nodes j such that arc $(i, j) \in A^k$. Similarly, $I(j, k)$ represents the nodes i such that $(i, j) \in A^k$. Now we can construct the set of nodes such that:

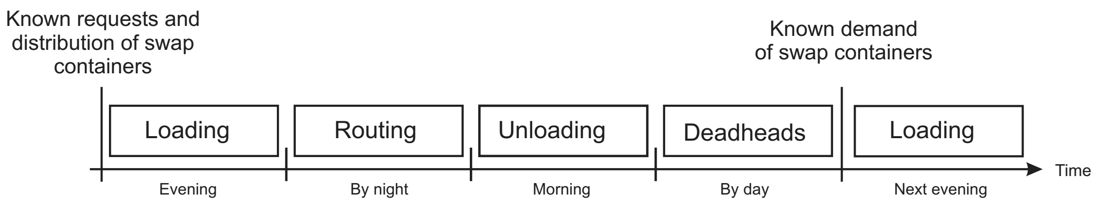

Fig. 1. The swap container problem in the course of time.