Heuristics for solving a capacitated multiple allocation hub location problem with application in German wagonload traffic

Julia Sender\textsuperscript{1,2}

\textit{Institute of Transport Logistics}
\textit{TU Dortmund}
\textit{Dortmund, Germany}

Uwe Clausen\textsuperscript{3}

\textit{Fraunhofer Institute for Material Flow and Logistics}
\textit{& Institute of Transport Logistics}
\textit{TU Dortmund}
\textit{Dortmund, Germany}

\section*{Abstract}
In this paper, we present a capacitated multiple allocation hub location problem, which arose from a network design problem in German wagonload traffic. We develop heuristic solution approaches based on local improvements. We solve the problem with the heuristics and CPLEX on test data sets provided by our partner Deutsche Bahn AG. The computational results are presented and compared.

\textit{Keywords:} hub location problems, IP, local search, logistics, network design
1 Introduction

Wagonload traffic is a specific production form in railway logistics, where the traffic volume (consisting of single wagons or small wagon groups) does not justify an entire direct train due to high fixed costs for running a train (e.g., track fees, personnel and traction costs). In order to exploit the effects of consolidation and economies of scale, the wagons are re-classified (bundled) in several formation yards with wagons of different origins and destinations. Although, the consolidation may lead to lower transportation costs, the reclassification of wagons in yards increases costs due to establishing and operating yards. Due to the strategic dimension of the network design, it has a great influence on future quality, production, and network costs of wagonload traffic. In order to route the origin-destination demand through the network, hub location problems take the location of hubs and the allocation of nodes into account. Thus, they decide about the structure of hub-and-spoke networks. Current reviews on hub location problems are given, e.g., in [1,2,3]. Most hub location problems can be divided into single and multiple allocation variants. In single allocation each non-hub node is allocated to a single hub node, whereas in multiple allocation each non-hub node may be allocated to multiple hub nodes. Multiple allocation versions increase the flexibility and complexity, but are expected to have lower costs. In most hub location problems the triangle inequality for costs or distances is assumed. Indeed, in wagonload traffic the costs per km may vary for different tracks. Since we consider a strategic problem, we introduce several capacity levels for dimensioning hubs. In wagonload traffic a shunting track in a yard is used to classify trains for a particular direction. Hence, the number of outgoing connections (directions) may also be limited for each yard. Thus, the considered capacitated multiple allocation hub location model differs in some ways from existing hub location problems. A similar model is also presented more in detail in our work [5], where we presented several modifications of the model in order to reduce the size and the complexity. These modifications include, e.g., the removal of redundant constraints, aggregation of constraints or adding of valid inequalities. However, the modifications done there have only small effects and therefore, are not sufficient to solve real-sized data sets with (commercial) solver, like CPLEX. In this paper, we develop heuristic solution approaches based on local

---

1 We thank the German Federal Ministry of Education and Research (BMBF) for supporting this research (03MS640B).
2 Email: julia.sender@tu-dortmund.de
3 Email: uwe.clausen@iml.fraunhofer.de
improvements for the original model.

In section 2, we present the optimization model. Solutions approaches for solving this model are developed in section 3. In section 4, computational results are given. We end with a short conclusion in section 5.

2 Optimization Model

Let $N$ be the set of origin-destination nodes and potential hub nodes and let $i, j, k, l, \in N$. The demand of origin-destination pair $(i, j)$ is given by a number of wagons $W_{ij}$. Since the network structure is mainly determined by the number of wagons which have to be consolidated in yards, the data set given does not contain origin-destination demand where an entire train, that is a direct connection, would be provided. Let $\Gamma_k = \{1, \ldots, s_k\}$ be the set of different capacity levels of hub $k$. The costs, capacity, and maximum number of outgoing connections of a hub $k$ with level $q$ ($q \in \Gamma_k$) are given by $c^q_k$, $\gamma^q_k$, and $b^q_k$. The costs per wagon on arc $(k, l)$ are represented by $c_{kl}$. We consider a discount factor $\beta$ for inter-hub transportation.

We introduce three different kinds of integer flow variables for collection, transfer, and distribution movements. The collection variables $X_{ik}$ represent the flow from origin $i$ to hub $k$. Transfer movements from hub $k$ to hub $l$, where the flow originates at origin $i$, are given by $Y_{kl}^i$. The distribution variables $Z_{lj}^i$ represent the flow from hub $l$ to destination $j$ originating at origin $i$. Moreover, we use binary variables for locating hubs and capacity levels. The location variable $H_k$ is one if node $k$ is a hub and zero otherwise. If node $k$ is a hub with capacity level $q$ ($q \in \Gamma_k$), the binary variable $H^q_k$ is one. To model the limitation on outgoing hub arcs, we introduce the binary variable $A_{kl}$ which is one if hub node $k$ is connected to node $l$ and zero otherwise. The capacitated multiple allocation hub location model is given as follows:

$$
\begin{align*}
\min & \sum_{k \in N} \sum_{q \in \Gamma_k} c^q_k H^q_k + \sum_{k \in N} \sum_{l \in N} c_{kl} (X_{kl}^i + \sum_{i \in N} (\beta Y_{kl}^i + Z_{lj}^i)) \\
\text{s.t.} & \sum_{i \in N} (X_{ik} + \sum_{l \in N} Y_{ik}^l) \leq \sum_{q \in \Gamma_k} \gamma^q_k H^q_k \quad \forall k \in N \\
& \sum_{q \in \Gamma_k} H^q_k = H_k \quad \forall k \in N \\
& Y_{kl}^i + Z_{lj}^i \leq O_i A_{kl} \quad \forall i, k, l \in N \\
& \sum_{l \in N} A_{kl} \leq \sum_{q \in \Gamma_k} b^q_k H^q_k \quad \forall k \in N
\end{align*}
$$
The objective function (1) minimizes the total costs, consisting of hub and transportation costs, where inter-hub flow is discounted. Constraints (2) assure that all flow into a hub may not exceed the capacity assigned to it. Exactly one capacity level has to be chosen for each established hub (cf. constraints (3)). In order to restrict the number of outgoing connections constraints (4), (5), and (6) are used. Constraints (7), (8), and (9) are flow related constraints. If an origin-destination node becomes a hub node, the demand has to be (re-)classified and thus, influences the capacity of the hub. Due to this, we assure that if a node becomes a hub, we have “artificial” collection or distribution movements from the hub to itself (cf. constraints (10) and (11)). Constraints (12)-(15) ensure usable solutions: a hub has to be established for each collection, transfer, and distribution movement. Constraints (16) are domain constraints. In account for practical requirements, it might be useful to assume a maximum distance between non-hub and hub nodes. This can be done by fixing corresponding collection and distribution variables to zero.

3 Local Search Heuristics

Heuristics based on local improvements are well known in hub location literature since they can provide good solutions without consuming a great amount of CPU time [1,2]. Due to the publicity of local search algorithms, we only describe the main details of the heuristic developed here. The approach exploits some ideas of possible moves of the simulated annealing approach in [4] for a (classical) capacitated hub location problem with single allocation.
One main element of applying local search approaches to specific problems is the definition of the neighborhood and, in particular, the definition of moves to get from one solution to another. We define different hub-based and flow-based moves to solve the problem which exploit the structure of the problem: location and allocation decisions. In some moves so called nearby non-hub or hub nodes of the current node are used. These nearby nodes are determined as follows: According to a sorted list (in terms of transportation costs), the best nodes are chosen to be nearby nodes of the given node. Additionally, a randomly chosen candidate is added. Finally, one nearby node is randomly chosen out of this list.

We consider the following hub-based moves: Relocate Hub exchanges the location of a hub with a nearby non-hub node. The corresponding flow is allocated to the new hub. The move New Hub establishes a new hub node; a non-hub node becomes a hub node. By doing so, the corresponding collection and distribution flow of the new hub is allocated to itself. Close Hub closes a hub node. The corresponding flow is allocated to a nearby hub node. The move Split Hub reallocates some origin-destination flows via a hub to an allocated non-hub node, which becomes a hub node. By doing so, the flow of a hub is split and a new hub is established.

Moreover, we consider some flow-based moves: By applying the move Relocate Flow Collection, the collection flow of an origin (non-hub) node to a hub node is reallocated to a nearby hub node. Similarly, Reallocate Flow Distribution changes the distribution hub of an origin-destination flow. Corresponding transfer and collection movements are modified. The move Swap Flows Collection chooses flows from two origin-destination pairs in the collection phase (with different, but nearby collection hubs) and swaps the allocation of the two collection flows. In other words, each flow is allocated to the hub of the other one. Similarly, the move Swap Flows Distribution reallocates the distribution flow of two origin-destination pairs to the nearby (distribution) hub of the other origin-destination pair. In both cases, affected movements are modified. By applying Shift Flow, an origin-destination flow is transported via an additional hub. Thus, for a chosen origin-destination pair a movement between two hubs is replaced by two new movements from the first hub to a (randomly) chosen hub and from the chosen hub to the second hub.

The main structure of our local search approach is given as follows. At first, we generate a feasible initial solution. According to a sorted list (in terms of transport volumes) an initial hub set is chosen, whereat the capacity of these hubs exceeds the total demand. Moreover, we add some supplementary hubs. Starting with this initial hub set, the solution is built up successively. For each
origin-destination pair collection, transfer, and distribution movements are
determined. If all origin-destination nodes are allocated, the initial heuristic
stops; otherwise another hub node is added to the hub list, all movements
are deleted, and the prior steps are repeated. In the following iterations, we
generate multiple neighbor solutions of the current solution. The best feasible
solution is compared to the current best solution. If the new solution is better
than the current best solution, we replace it. Infeasible solutions created by
the moves are not considered as starting points for further search. In order to
generate a new solution in the neighborhood of the current one, we choose one
of the moves described above. The (preliminary chosen) probabilities that one
move is chosen are set to 15%, 5%, 5%, 5%, 15%, 15%, 10%, 10%, and 20%,
respectively. The setting is based on previous tests and the results of [4].

We implemented three versions of the local search approach described
above. In order to escape from local minima, we implemented a local search
with multistarts MLS and a simulated approach SA. In MLS we start the local
search procedure several times to escape from local optima. This is done by
applying moves to the current best solutions and accepting all feasible solu-
tions for a given number of iterations. Moreover, we accept a new solution
for the further search if it is at most 10 percent worse than the current best
solution. This is a version of threshold accepting, i.e. accepting worse solu-
tions within a given threshold. Based on the same input parameters, we also
implemented a simulated annealing version, where the probability to accept
worse solutions decreases with an increasing run-time. Solutions increasing
the objective value are accepted according to the Boltzmann’s probability
(depending on the decreasing temperature). The two heuristics are also com-
pared to a pure local search (descent) variant DLS. While the heuristic MLS and
SA allow uphill moves, the descent approach accepts only improvements. The
stop criterion is given by a time limit (here 43,200 sec) or by a given number
of iterations (here 500 iterations) with no improvement of the objective value.
Moreover, we test a simple constructive heuristic CH. This heuristic is similar
to the heuristic used for generating the initial solution, whereat we do not add
additional hub nodes.

4 Computational Results

All heuristics were implemented in MATLAB R2011b. In order to compare
the quality of the heuristic solutions, the model was implemented in GAMS
23.9.1 and solved with CPLEX 12.4.0.1. All tests were carried out on a PC
under Windows 7 (64 bit) with an Intel Core i5 CPU, 3.2 GHz, and 16 GB
RAM. Aiming at an approach applicable in practice, the time limit was set to 12 hours.

In the computational experiments we test the effectiveness of the heuristics with different data sets provided by Deutsche Bahn AG. The results are presented in Table 1. The columns gap represent the relative gap. According to CPLEX, we define the gap as \( \text{gap} = 100 \cdot \frac{BF - BP}{BF} \), where we use the lower bounds \((BP)\) found by GAMS/CPLEX and the best feasible solution \((BF)\) found by the corresponding solution approach. Since \text{DLS} and \text{CH} do not reach the time limit of 12 hours, we also present the time used. Since the random elements have an influence on the results, the heuristics are repeated five times for each data set. The average gap is based on the average \(BF\); the presented deviations are based on the original \(BF\) in comparison to the average \(BF\).

<table>
<thead>
<tr>
<th>Instance</th>
<th>GAMS/CPLEX</th>
<th>MLS</th>
<th>SA</th>
<th>DLS</th>
<th>CH</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(\text{no. nodes})</td>
<td>(\text{gap (%)})</td>
<td>(\phi \text{ gap (%)})</td>
<td>(\phi \text{ gap (%)})</td>
<td>(\phi \text{ gap (%)})</td>
</tr>
<tr>
<td>1</td>
<td>37</td>
<td>13.98</td>
<td>12.20 ((\pm3.20%))</td>
<td>14.47 ((\pm3.66%))</td>
<td>16.53 ((\pm3.45%))</td>
</tr>
<tr>
<td>2</td>
<td>49</td>
<td>21.59</td>
<td>15.70 ((\pm3.65%))</td>
<td>21.87 ((\pm2.32%))</td>
<td>23.22 ((\pm4.20%))</td>
</tr>
<tr>
<td>3</td>
<td>74</td>
<td>-</td>
<td>20.77 ((\pm5.81%))</td>
<td>26.43 ((\pm2.82%))</td>
<td>26.32 ((\pm2.54%))</td>
</tr>
<tr>
<td>4</td>
<td>74</td>
<td>4.60</td>
<td>9.97 ((\pm1.21%))</td>
<td>12.23 ((\pm1.41%))</td>
<td>21.24 ((\pm7.04%))</td>
</tr>
</tbody>
</table>

Table 1
Computational results

Even for small test data sets no optimal solutions are found by CPLEX within a run-time of 12 hours. Although, no heuristic can find (proven) optimal solutions, the results in Table 1 are promising. In most cases tested, MLS can outperform the results found by CPLEX – even for small scenarios. By comparing the heuristic approaches, we observe that MLS outperforms the other heuristics. In particular, the solutions found by MLS outperform the best solutions found with heuristics SA and DLS. Although, the simulated annealing approach accepts solutions worse than the current best solutions, we observe once ”good” best solutions are found, the heuristic cannot improve these values (for hours). We observed this effect for varying input parameters for the acceptance criterion, i.e., the initial temperature and the annealing schedule. One reason might be that the defined neighborhoods are not appropriate for simulated annealing. Due to the (random) multistart, which accepts all feasible solutions for a given numbers of iterations, we get a greater chance to escape from local optima. In particular, we observe that the limitation of outgoing connections increases the difficulty to find feasible (and better) solutions in the neighborhood of a solution. By considering the solution process, we noticed that satisfying solutions are found within a small amount of run
time and hence, can be used as upper bounds within in an exact algorithm.

5 Conclusion and future work

In this paper, we developed different versions of a local search approach for solving a capacitated multiple allocation hub location problem derived from a practical application in network design of German wagonload traffic. These heuristics are based on local improvements and exploit the structure of the problem: location and allocation decisions. First computational results, in particular of the local search with multistart, are promising for solving real data sets. All heuristic approaches can be used to find satisfying solutions in a small amount of run time. From a practical point of view, these heuristics can be easily extended to other practical requirements, like vehicle-based costs, due to the modular construction. Although, first computational results are promising, in future work we consider further tests on the parameters, e.g. probabilities, in order to improve the results. Moreover, future work is dedicated to improving the heuristics in terms of the final solution, i.e. closing the gap, as well as solving larger hub location problems.

References


