The Uncapacitated Single Allocation p-Hub Median Problem with Stepwise Cost Function

Borzou Rostami[a], Christoph Buchheim [b]

[a] *Department of Mathematics and Industrial Engineering, Polytechnique Montreal and GERAD, Montréal, Canada*

[b] *Fakultät für Mathematik, TU Dortmund University, Germany*

Abstract

We address a new version of the Uncapacitated Single Allocation p-Hub Median Problem (USApHMP) in which transportation costs on each arc are given by piecewise constant cost functions. In the classical USApHMP, transportation costs are modelled as linear functions of the transport volume, where a fixed discount factor on hub-hub connections is introduced to simulate economies of scale. We argue that, in many cases, it is more realistic to replace the discount factor by decision variables representing the number of vehicles required on the given arc. Introducing integral variables for vehicle numbers makes the resulting optimization problem significantly harder, especially for the existing state-of-the-art MILP solvers. We propose an exact branch-and-bound algorithm where lower bounds are computed by considering a Lagrangian relaxation procedure. The Lagrangian relaxation exploits the problem structure and decomposes the problem into a set of smaller subproblems that can be solved efficiently. In particular, some of the latter can be reduced to continuous knapsack problems that can be solved quickly. We report exact solutions for instances with up to 50 nodes, i.e., significantly larger instances than those solvable by MILP solvers.

Keywords: Hub location, Economies of scale, Lagrangian relaxation, Branch and Bound.

1. Introduction

We consider transportation networks modelled by complete graphs $G = (V, E)$, where the node set $V = \{1, 2, \ldots, n\}$ represents the origins, destinations and possible hub locations, and E is the edge set. Let W_{ij} be the total amount that should be sent from node i to node j (often called the "flow"). Since it can become very expensive to transport all shipments directly, it is often advisable to establish a number of *hubs* to receive the shipments from several origins and redirect them to the destinations. In other words, the hub nodes are used as transhipment and consolidation nodes. The main idea behind this is *economies of scale*: While the construction and operation of hubs and the resulting detours lead to extra costs, the bundling of shipments decreases costs.

Once the hub nodes are chosen, the non-hub nodes are allocated to them. Depending on the allocation strategy of assigning the non-hub nodes to the hub nodes, there are two different classes of hub location problems: multiple allocation hub location problem and

single allocation hub location problem. In the former, a non-hub node can be allocated to more than one hub of the network; this approach has applications in aviation industry, where the main goal is to exploit the economies of scale obtained by using larger aircraft between hubs. In the latter, every non-hub node is allocated to exactly one hub node; this approach is applied in situations in which sorting at the source is not possible (or too costly) so that all shipments are transported as a whole to the allocated hub.

The first mathematical model for the hub location problem was given by O'Kelly [1]. He proposed a quadratic integer programming formulation for the problem of minimizing the total transport cost for a given number of hubs to locate (p-Hub Median Problem). The following hub location literature mainly focused on the transportation cost objective locating both a fixed and a variable number of hubs. Important early studies are [2, 3, 4, 5], where different kinds of hub and capacity constraints were discussed. On the heuristic side, recent successes are a neighbourhood search algorithm of Ilic et al. [6], a discrete particle swarm optimization of Bailey et al. [7] and a genetic algorithm by Topcuoglu et al. [8]. Contreras et al. contributed much to the field: In [9], a Lagrangian relaxation approach is developed, whereas [10] considers Benders decomposition and [11] uses a branch-and-price approach. For a more complete and also historical introduction, the reader is referred to [12, 13].

The large majority of literature models the economies of scale as a linear function of the transport volume multiplied by a fixed constant α on each hub-hub connection. Estimating α a priori can be difficult, and any fixed factor can only be a rough estimate of the real savings. It seems appropriate to look for other models of transportation costs which represent the real transportation costs more appropriately. Two main approaches are found in the literature:

Concave cost functions: Decreasing unit costs for increasing transport volumes can be approximated by a concave cost function. This cost function is usually chosen as piecewise linear function which can be justified by practical (a finite number of possible modes of transport) or model-theoretic reasons (because linearity eases solvability). Cunha and Silva [14] solve a hub location problem with concave piecewise linear function by a genetic algorithm. de Camargo et al. [15] use a concave, piecewise linear function on hub-hub-connections and linear costs for other connections (FLOWLOC model) and solve the problem by Benders' decomposition algorithm.

Vehicle-based costs: The costs for trucks, trains and planes mainly depend on the travelled distance while the filling quota plays a minor role. Hence it is sensible to measure costs *per vehicle* and not *per volume*. Song [16] was one of the first to define a vehicle-based hub location problem. He modelled the costs a step function for a problem in air transport. The work of van de Leensel [17] used a similar model for discrete capacities in a telecommunication network. In [18] Sender and Clausen present a heuristic approach to a similar problem in rail freight. Meier [19, 20] considered heuristics for LTL problems with multiple allocation pattern. Generally, as also noted by [21], these problems become very difficult MIP because the economies of scale are purely modelled by the integrality of the number of vehicles; the relaxed LP version has no sense of bundling and gives very weak lower bounds. Successful approaches with standard MIP solvers are only documented for simplified problems [22, 23].

In this paper we consider a variant of the *Uncapacitated Single Allocation p-Hub Median Problem* (USApHMP) in which the cost for transport is measured by the number of vehicles

needed on each connection. The USApHMP is the original formulation of O'Kelly [1], recent successful approaches with heuristics [6] and exact methods [24, 25] exist. Our version of the problem in which the vehicles are included directly is called *Uncapacitated Single Allocation p-Hub Median Problem with vehicle sizes* (USApHMPv) and arises in postal applications where connections are usually serviced by (large) vehicles. The capacity utilization of these vehicles is the main factor in successful consolidation. Using vehicle sizes leads to a stepwise cost function, where the step width is the capacity of the vehicle. However, our vehicle-based approach can also be combined with a component of the cost function that is linear in the transport volume.

Such stepwise cost functions make the problem much harder to solve. They have been considered also by Yaman and Carello [26] for a problem arising in the design of telecommunication networks, where the stepwise function results from the integrality of the number of links on a hub-hub-connection. A branch-and-cut approach is devised to solve the problem.

We propose a different solution approach for the USApHMPv. Starting from a natural quadratic mixed integer program, we first provide a linearized model. The introduction of vehicle variables shifts the quadratic term from the objective function in the USApHMP to a set of constraints; the costs are determined per vehicle, and the number of vehicles on each connection is determined by the transport volume. Hence the difficulty of the resulting problem does not only stem from its quadratic nature, but also from the additional integrality requirement. The proposed MILP formulation is based on the four-index formulation of Skorin-Kapov et al. [5] for the USApHMP. The main difficulties in solving the continuous relaxations of these MILPs are the huge number of variables and constraints that appear in the model, moreover the LP relaxations yield very weak dual bounds in general. Therefore, we develop a Lagrangian relaxation scheme to obtain tight upper and lower bounds. The Lagrangian relaxation exploits the problem structure and decomposes the problem into a set of smaller subproblems that can be solved efficiently. Some of the latter can be reduced to continuous knapsack problems that can be solved quickly. It is crucial that we maintain integrality requirements in these subproblems. Due to this, the proposed Lagrangian relaxation does not have the so-called integrality property, so that the obtained bound will be stronger than the one given by the continuous relaxation of the proposed MILPs. The Lagrangian relaxation is then embedded into a branch-and-bound scheme in order to solve the problem to optimality.

The paper is organized as follows: Section 2 describes the quadratic formulation of the problem and its linearized model. Section 3 presents a method to obtain strong lower bounds by Lagrangian relaxation. Section 4 explains how to embed the lower bounding methods into a branch-and-bound framework. Section 5 lists and interprets computational results obtained with our approach.

2. Problem formulation and linearizations

In this section we define and discuss a vehicle-based version of the USApHMP. To this end, we keep the constraints of the USApHMP but use a different method to compute the transportation cost. Before formulating the problem, let us first introduce the following

notation for a given network.

$V = \{1, 2, \ldots, n\}$	Set of nodes.
W_{ij}	Volume of the shipment to be transported from i to j, for $i, j \in V$.
q_{ij}	Capacity of a vehicle on arc (i, j) for all $i, j \in V$.
d_{ij}	Cost of a vehicle on arc (i, j) for all $i, j \in V$.
$O_i = \sum_{j \in V} W_{ij}$	Total outgoing volume from node i.
$D_i = \sum_{j \in V} W_{ji}$	Total incoming volume to node i.

2.1. Preprocessing

In order to formulate the problem in an efficient way, we perform a preprocessing step for computing the depot-hub and hub-depot transportation costs. The USApHMPv uses the *single allocation pattern*. As the total incoming and outgoing volume of each depot i are given (namely D_i and O_i, respectively), this implies directly that we can precompute cost factors c_{ik} for connecting depot i to hub k. This holds for any kind of cost function. In particular, in the vehicle-based setting, we obtain

$$c_{ik} = d_{ik}\, q_{ik}^{-} + d_{ki}\, q_{ik}^{+} \,,$$

where $q_{ik}^{-} = \lceil O_i / q_{ik} \rceil$ is the required number of vehicles for transport from node i to node k and $q_{ik}^{+} = \lceil D_i / q_{ik} \rceil$ is the required number of vehicles for transport from node k to node i.

This implies that integral variables for vehicles on depot-hub and hub-depot connections are superfluous. Therefore (as we will see later), we only explicitly introduce vehicle variables for hub-hub connections.

2.2. Quadratic binary formulation

The USApHMPv consists in selecting p nodes as hubs and assigning the remaining nodes to these p hubs such that each non-hub node is assigned to exactly one hub node with the minimum overall transportation cost, as specified below. We define binary variables x_{ik} indicating whether a source/sink i is allocated to a hub k. The variables x_{kk} are used to indicate whether k becomes a hub. Moreover, we define an integer variable v_{km} to be the necessary number of vehicles on the hub-hub arc (k, m). The USApHMPv can then be formulated as follows, where the objective function measures the total transport costs consisting of the collection and distribution costs of nonhub-hub and hub-nonhub connections as well as the hub-hub vehicle costs.

$$
\begin{aligned}
\text{USApHMPv:} \quad \min \quad & \textstyle\sum_i \sum_k c_{ik}\, x_{ik} + \sum_k \sum_{m \neq k} d_{km}\, v_{km} & & \\
\text{s.t.} \quad & \textstyle\sum_k x_{kk} = p & & (1)\\
& \textstyle\sum_k x_{ik} = 1 & \forall i \in V \quad & (2)\\
& x_{ik} \leq x_{kk} & \forall i,k \in V \quad & (3)\\
& \textstyle\sum_i \sum_j W_{ij}\, x_{ik} x_{jm} \leq q_{km}\, v_{km} & \forall k,m \in V, k \neq m \quad & (4)\\
& x_{ik} \in \{0,1\} & \forall i,k \in V \quad & (5)\\
& v_{km} \geq 0, \text{ integer} & \forall k,m \in V, k \neq m \quad & (6)
\end{aligned}
$$

Equation (1) makes sure that exactly p hubs are chosen. Constraints (2) force every source/sink to be allocated to precisely one hub node. Constraints (3) state that i can only be allocated to k if k is chosen as a hub. Constraints (4) relate the number of the vehicles on each hub-hub connection to the total flow passing the link and state that the flow on each hub-hub connection cannot exceed the capacity of vehicles assigned to it.

In the above model, if the vehicle capacities q_{km} and costs d_{km} tend to zero (with the same rate), the integrality constraint on v_{km} disappears and the problem converges to the traditional USApHMP with a discount factor of $\alpha_{km} = 1/q_{km}$ on the hub-hub connection (k,m). The discount factor of the classical formulation is independent of k and m, but this has no influence on the model structure. However, as long as only a moderate number of vehicles is needed, the integrality requirement for v_{km} has a significant impact on the resulting solution (and the difficulty of solving the problem).

Note that our model can easily be combined with the traditional model using linear transportation costs by adding an appropriate quadratic term of the form $\sum_{i,j,k,m} \alpha W_{ij} b_{km} x_{ik} x_{jm}$ in the objective function, where b_{km} is the cost per unit of flow between nodes k and m ($k,m \in V$), and α is discount factor, i.e., the total costs are given as a sum of vehicle-based and volume-based costs. This term has no influence on the applied Lagrangian relaxation (as the reader will easily see later), but is left out to simplify the discussion.

2.3. Linearizations

Due to the quadratic structure of USApHMPv, the standard approach for solving the problem is to linearize it in order to obtain a mixed integer linear program (MILP). The continuous relaxation of the MILP is then solved to optimality to compute a lower bound. Therefore, the first step to obtain a lower bound is to linearize the quadratic terms $x_{ik}x_{jm}$ for all $i,k,j,m \in V$ using the standard linearization technique based on the introduction of new non-negative binary variables $y_{ikjm} = x_{ik}x_{jm}$ which satisfy the following set of constraints:

$$y_{ikjm} \leq x_{ik}, \quad y_{ikjm} \leq x_{jm}, \quad \text{and} \quad y_{ikjm} \geq x_{ik} + x_{jm} - 1.$$

The main disadvantage of the standard linearization is that it does not take the problem structure into account and thus leads to very weak dual bounds in general after relaxing

integrality. To overcome this problem, we follow the classical hub location linearization of Skorin-Kapov et al. [5] and introduce binary variables y_{ikjm} for all $i, k, j, m \in V$. Here y_{ikjm} is 1 if the flow from node i to node j travels via hubs located at nodes k and m. The resulting formulation is as follows:

$$
\begin{aligned}
\text{LP4:} \quad \min \quad & \textstyle\sum_i \sum_k c_{ik} x_{ik} + \sum_k \sum_{m \neq k} d_{km}\, v_{km} & & \\
\text{s.t.} \quad & \textstyle\sum_i \sum_{j \neq i} W_{ij}\, y_{ikjm} \leq q_{km} v_{km} & \forall k, m \in V, k \neq m & (7) \\
& \textstyle\sum_k y_{ikjm} = x_{jm} & \forall i, j, m \in V, i \neq j & (8) \\
& \textstyle\sum_m y_{ikjm} = x_{ik} & \forall i, j, k \in V, i \neq j & (9) \\
& 0 \leq y_{ikjm} \leq 1 & \forall i, k, j, m \in V, i \neq j & (10) \\
& (1), (2), (3), (5), (6) & &
\end{aligned}
$$

where constraints (8) and (9) enforce $y_{ikjm} = x_{ik} x_{jm}$. The integrality of y_{ikjm} is implied by the integrality of x_{ik} and (8), (9). Note that USApHMPv and LP4 are equivalent in that for any feasible solution to one problem there exists a feasible solution to the other problem with the same objective function value. However, if the integrality restrictions on variables x or v are relaxed, the formulation LP4 is no longer equivalent to USApHMPv and it provides a lower bound on the objective function value.

Ernst and Krishnamoorthy [3] proposed a set of 3-index flow variables for linearizing the USApHMP, which led to models that could achieve a better tradeoff between compactness and tightness. In the USApHMP the actual costs between nodes satisfy the triangle inequality which ensures that each routed flow from each origin will pass through at most two hub nodes. However, this 3-index flow formulation is not valid for USApHMPv, as the costs on the hub-hub connections in USApHMPv are based on integer numbers of vehicles.

3. Bounds based on a Lagrangian relaxation

The introduction of variables v counting vehicles shifts the quadratic term from the objective function in the USApHMP to a set of constraints in the USApHMPv; the costs are determined per vehicle, and the number of vehicles on each connection is determined by the transport volume. Therefore, the usual USApHMP can be viewed as a relaxation of a vehicle based cost function USApHMPv. Hence the difficulty of the USApHMPv does not only stem from its quadratic nature, but also from the additional integrality requirement on vehicle variables v. Relaxing the integrality of variables x and v in the linearized model LP4 of Section 2 provides lower bounds for the USApHMPv. However, there are two main challenges: First, due to the large number of variables and constraints, solving the continuous relaxation of LP4 requires a considerable CPU time as the size of the instances increases. Second, relaxing the integrality requirement on the vehicle variables v leads to large duality gaps, as also observed in our experimental results; see Section 5. These observations motivate us to study a stronger lower bounding approach based on a Lagrangian relaxation that fully exploits the problem structure.

The idea of Lagrangian relaxation is to transfer one or more complicating constraints to the objective function, so that the resulting problem splits into a set of easy problems. Let us apply a Lagrangian relaxation to constraints (2), (8), and (9) of the LP4 formulation using a set of Lagrangian multipliers π_i for $i \in V$, λ_{ijm} for $(i,j,m) \in V^3$ with $i \neq j$, and μ_{jik} for $(j,i,k) \in V^3$ with $i \neq j$, respectively. The resulting Lagrangian function is as follows:

$$\begin{aligned} \mathcal{L}(\pi,\lambda,\mu) = \min \quad & K + \sum_i \sum_k \bar{c}_{ik} x_{ik} + \sum_k \sum_{m \neq k} d_{km}\, v_{km} + \sum_i \sum_{j \neq i} \sum_k \sum_m \bar{f}_{ikjm} y_{ikjm} \\ \text{s.t.} \quad & (1), (3), (5), (6), (7), (10) \end{aligned}$$

where

$$\begin{aligned} & K = \textstyle\sum_i \pi_i \\ & \bar{c}_{ik} = c_{ik} - \pi_i - \textstyle\sum_{j \neq i} (\lambda_{jik} + \mu_{jik}) && \forall i,k \in V \\ & \bar{f}_{ikjm} = \lambda_{ijm} + \mu_{jik} && \forall i,j,k,m \in V, i \neq j \end{aligned}$$

The best lower bound is then obtained by solving the following Lagrangian dual problem:

$$\max_{\pi,\lambda,\mu} \mathcal{L}(\pi,\lambda,\mu)$$

Due to the choice of the relaxed constraints, the problem decomposes, so that solving the Lagrangian function reduces to solving each of the following subproblems independently:

$$\mathcal{L}_x := \min\Big\{ \sum_i \sum_k \bar{c}_{ik} x_{ik} : \ (1), (3), (5) \Big\} \tag{11}$$

$$\mathcal{L}_{vy} := \min\Big\{ \sum_k \sum_{m \neq k} d_{km}\, v_{km} + \sum_i \sum_{j \neq i} \sum_k \sum_{m \neq k} \bar{f}_{ikjm} y_{ikjm} : \ (6), (7), (10) \Big\} \tag{12}$$

$$\mathcal{L}_y := \min\Big\{ \sum_i \sum_{j \neq i} \sum_k \bar{f}_{ikjk} y_{ikjk} : \ (10) \Big\} \tag{13}$$

Observe that Subproblem (11) can be solved by a simple inspection: Define, for each $k \in V$,

$$r_k = \bar{c}_{kk} + \sum_{i \neq k} \min\{0, \bar{c}_{ik}\}, \tag{14}$$

then Subproblem (11) is reduced to selecting the p hubs with smallest cost r_k. Afterwards, x_{ik}, $i \neq k$, is set to one if and only if k is a hub and $\bar{c}_{ik} < 0$. Subproblem (13) has to be included into the list because Subproblem (12) only contains the variables y_{ikjm} with $k \neq m$. It is trivial because no constraints apply except for $0 \leq y_{ikjk} \leq 1$. Therefore, it is only necessary to check whether $\bar{f}_{ikjk}$ is positive or negative.

It thus remains to solve Subproblem (12), which requires a more sophisticated approach, described in the remainder of this section. First note that Subproblem (12) can be further decomposed into $n(n-1)$ subproblems, one for each (k,m) with $k \neq m$ as follows:

$$\begin{aligned}
\max \quad & -d_{km}\, v_{km} - \sum_i \sum_{j \neq i} \bar{f}_{ikjm} y_{ikjm} \\
\text{s.t.} \quad & \sum_i \sum_{j \neq i} W_{ij}\, y_{ikjm} \leq q_{km}\, v_{km} \\
& 0 \leq y_{ikjm} \leq 1 && \forall i, j \in V, i \neq j \\
& v_{km} \geq 0, \text{ integer.}
\end{aligned}$$

Here we turned the minimization problem into a maximization problem, as we will exploit its connection to a continuous knapsack problem. In fact, each of these subproblems can be written in the following form, which we call Vehicle Continuous Knapsack Problem (VCKP):

$$\begin{aligned}
\max \quad & -d\, v - \sum_{l \in L} \bar{f}_l y_l \\
\text{s.t.} \quad & \sum_{l \in L} W_l\, y_l \leq q\, v && (15) \\
& 0 \leq y_l \leq 1 && \forall l \in L \quad (16) \\
& v \geq 0, \text{ integer} && (17)
\end{aligned}$$

where $\bar{f}_l < 0$ for all $l \in L$. Here, we left out the indices k and m and subsumed the index pairs (i,j) as l. Variables with positive $\bar{f}_l$ were eliminated because they are necessarily zero.

For fixed v, the problem is a Continuous Knapsack Problem (CKP), which is the LP-relaxation of the classical Knapsack Problem. It would thus be possible to solve VCKP by solving the stated CKP for $v = 0, 1, 2, \ldots$ until v reaches a precomputed maximum. But the following theorem shows that there are at most two values of v which have to be considered to find an optimal solution of VCKP.

Theorem 1. *Define $\Omega = \{l \in L \mid \bar{f}_l \leq -dW_l/q\}$ and $W_\Omega = \sum_{l \in \Omega} W_l$. Then there exists an optimal solution of the VCKP such that v attains one of the rounded values $\lfloor W_\Omega/q \rfloor$ or $\lceil W_\Omega/q \rceil$.*

Proof. Assume that variables are sorted by non-increasing $-\bar{f}_l/W_l$ and $\Omega = \{l \in L \mid l \leq l^*\}$. Define the function $\phi\colon \mathbb{R}_{\geq 0} \to \mathbb{R}$ by

$$\phi(v) := \max \Big\{ -d\, v - \sum_{l \in L} \bar{f}_l y_l \Big| \sum_{l \in L} W_l\, y_l \leq q\, v;\ 0 \leq y_l \leq 1\ \forall l \in L \Big\}.$$

It is easy to see that ϕ is concave. The right derivative of ϕ at W_Ω/b is $-d - q\bar{f}_{l^*+1}/W_{l^*+1} < 0$, as y_{l^*+1} will be the variable increased next and $l^*+1 \notin \Omega$, while the left derivative at W_Ω/q is $-d - q\bar{f}_{l^*}/W_{l^*} \geq 0$ as $l^* \in \Omega$. Hence W_Ω/q is a (continuous) maximizer of ϕ. By concavity, the integer maximizer of ϕ is W_Ω/q rounded up or down. □

This implies that every VCKP can be solved by reduction to two CKPs, obtained by fixing v to each value $\lfloor W_\Omega/q \rfloor$ and $\lceil W_\Omega/q \rceil$, and by comparing the results.

The classical solution method of Dantzig [27] for the CKP is to first sort the items according to non-increasing profit-over-weight ratios $-\bar{f}_l/W_l$. Then an optimal solution of CKP is given by $y_l = 1$ for all l in this order until the volume exceeds bv or all elements of the list are used up. The last y_l may be fractional. Because the maximal number of items is $|V|^2$, this algorithm needs $O(|V|^2 \log |V|^2) = O(|V|^2 \log |V|)$ time in the worst case (sorting is the dominant factor). Following Korte et al. [28], the CKP can even be solved in linear time (in the number of items) by the weighted median algorithm. For the computational experiments, we implemented Dantzig's algorithm because the large majority of the CKP is expected to be quite small and the weighted median algorithm has a large overhead for separating and recursively solving subproblems.

The different VCKPs are independent and can be solved in parallel. This will be exploited for the numerical experiments discussed in Section 5.

4. Branch-and-bound algorithm

The Lagrangian relaxation approach described in the previous section will be used as the basis of an exact solution method to the problem. For that, we construct a subgradient algorithm for the Lagrangian relaxation and use the subgradient algorithm in a branch-and-bound framework. While this construction is largely standardized, there is a number of points to define and discuss.

First of all, the branching rules have to be defined. As we are working with binary variables, branching usually means to choose a variable and set it to zero or one in the resulting two branches. The method for choosing a variable will be detailed below in subsection 4.3.

Furthermore, the fixing of variables has to be exploited in the Lagrangian relaxation, increasing its strength. If all variables are fixed, the Lagrangian relaxation should lead to a feasible primal solution. Therefore, a complete exploration of the branch-and-bound tree leads to a provably optimal solution. The effects of fixing variables will be the topic of the subsection 4.2.

Lastly, a primal heuristic is specified, so that it is not necessary to branch down to the very bottom of the search tree to find feasible solutions. This is done in the following subsection.

Note that the subgradient algorithms of all nodes but the root node can be "warm-started": It can inherit the values of π, μ and ν from its parent node, so that the subgradient algorithm terminates much faster. Furthermore, if a subgradient algorithm exceeds the value of the (globally) best known upper bound, it can exit immediately.

4.1. Primal heuristic

In each iteration of the subgradient algorithm, we use the solution of the Lagrangian relaxation to compute a primal solution. Let $S = \{k_1, k_2, \ldots, k_p\}$ be the set of the p selected hubs in the solution of the Lagrangian relaxation. Then, for each $i \in V$, we determine the

subset $S_i \subseteq S$ of all hubs $k \in S$ that satisfy $x_{ik} = 1$. If $S_i \neq \emptyset$, we select the hub k^* with the minimum cost c_{ik^*}, i.e.,

$$k^* \in \operatorname{argmin}\Big\{c_{ik} : k \in S_i\Big\},$$

and set $x_{ik^*} = 1$. Otherwise, if $S_i = \emptyset$, we assign a random hub from the whole set S. In this way, we determine the values of all x variables for the primal solution. The values of y are computed as $y_{ikjm} = x_{ik}x_{jm}$, while the variable v_{km} is set to the lowest possible value according to constraint (7).

4.2. Fixing variables

As the subgradient method consists in iteratively solving the Lagrangian relaxation for different values of π, λ and μ, we need to adapt the Lagrangian function and solution method to the presence of fixed variables. We should note here that our branching strategies are based only on fixing the x variables because fixing all x variables automatically fixes all y variables by equations (8) and (9) (in fact it follows that $y_{ikjm} = x_{ik}x_{jm}$). This particularly means that fixing all x variables fixes all variables and the solution of the Lagrangian algorithm just returns the fixed variables, i.e. a feasible solution.

First of all, we assume that the fixed values of x are *implication complete*: This means that every non-fixed value of x can be set to zero or one without violating a constraint of the (unrelaxed) problem. For example, $x_{ik} = 1$ necessarily implies $x_{kk} = 1$, so that x_{kk} needs to be fixed to make the fixing of x implication complete. Furthermore, the fixing of the x variables has implications for the y variables: y_{ikjm} is fixed to zero if x_{ik} or x_{jm} are fixed to zero, and y_{ikjm} is fixed to one if both x_{ik} and x_{jm} are fixed to one. In the following we present the implications for the three subproblems $\mathcal{L}_x$, $\mathcal{L}_{vy}$, and $\mathcal{L}_y$ introduced in Section 3.

Let us consider Subproblem $\mathcal{L}_x$ expressed in (11). Suppose that p' is the number of variables that have been positively identified as hubs. Then we only need to choose $p - p'$ hubs from the unfixed hub variables. Moreover, in the calculation of q_k in (14), we need to eliminate all pairs (i, k) for which x_{ik} have been already fixed (to zero or one).

As for Subproblem $\mathcal{L}_{vy}$ presented in (12), all variables y_{ikjm} that have been already fixed to zero are eliminated from the model (just as if their coefficient was positive). Moreover, the variables y_{ikjm} fixed to one are also eliminated, but their elimination leads to a shift in the objective function – which does not create any difficulties – and in the constraint (15):

$$\sigma + \sum_l W_l\, y_l \leq q\, v,$$

where $\sigma \in \mathbb{R}$ is a positive constant. Note that Theorem 1 still holds for the solution of the above adaptation of VCKP, with one subtle difference: The solution for $\lfloor W_\Omega/q \rfloor$ may be infeasible. This is exactly the case if $q\,\lfloor W_\Omega/q \rfloor < \sigma$. In this case, we thus have to choose the solution for $\lceil W_\Omega/q \rceil$.

Subproblem $\mathcal{L}_y$ in (13) stays unaltered except for the fixed values of y_{ikjm}.

4.3. Branching rules

First of all, we decide to branch on the hub variables x_{kk} as long as one of these is unfixed. The reason is that the hub variables have the strongest influence on the solutions. To find the appropriate index k, we use the final Lagrangian multipliers (π, λ, μ) of the given branch-and-bound node. Let us define

$$B_k = \sum_i \sum_j \left(|\lambda_{ijk}|^2 + |\mu_{ijk}|^2 \right)$$

for every unfixed k. This is a measure of the deviation of the constraints (8) and (9) to be violated. We then branch on the variable x_{kk} with maximal B_k, since the multipliers suggest that the constraints (8) and (9) involving this index k are particularly relevant. If all hub variables x_{kk} are fixed, we analogously define

$$B_{ik} = \sum_j \left(|\lambda_{jik}|^2 + |\mu_{jik}|^2 \right)$$

for each unfixed x_{ik} and branch on the variable x_{ik} with largest value B_{ik}.

5. Computational experiments

In this section we present our computational experiments on the lower bound computations and the branch-and-bound algorithm. We evaluate the quality of lower bounds obtained from the linearized model LP4 as well as the time needed to compute them, and compare both with the Lagrangian relaxation based approach that we introduced.

For our experiments, we used a 3.4 GHz processor with 16GB RAM and four cores. We coded our algorithm in C♯. The LP4 formulation is solved by Gurobi 6.0. To obtain the optimal (near-optimal) Lagrangian multipliers Γ we use the subgradient optimization technique [29]. At the root node, 10000 iterations of the subgradient method are performed. This number was chosen because preliminary experiments showed that after 10000 iterations, the improvement of the lower bound stops or becomes very slow. The usual parameter that controls the step size in subgradient methods is initialized at value 1 and then halved whenever 100 iterations have passed without improving the best lower bound. For non-root nodes, we run the subgradient method for at most 100 iterations, as it already starts with a near-optimal value for the Lagrange multipliers inherited from the parent node. In the following, we first present the test instances and then provide the experimental results.

5.1. Test instances

For numerical tests, we use the Australian Post data set (AP), which can be found in the OR library [30] and which is used frequently for different hub location problems. We use instances from 10 to 50 nodes and vary the number of hubs p from 2 to 5.

Two pieces of data are missing: The vehicle cost d_{km} and the vehicle size q_{km}. We will derive d_{km} from q_{km} by setting $d_{km} = \alpha b_{km} q_{km}$, which is reasonable as αb_{km} is a per-volume

cost factor. Different choices of vehicle sizes q_{km} have to be tested to see the influence of our new model structure.

We used a fixed vehicle size q for each $k, m \in V$ because companies usually use only one kind of large vehicle. We use the next paragraph to get a rough estimate of usable values of q. The assumptions that are made to get this estimate are not part of the model or the computational experiments.

Very small values for q lead to a model which is nearly equivalent to USApHMP, so that these are covered by the traditional approaches. We aim for situations in which each hub-hub connection is served by about 1 to 5 vehicles. Let $W = \sum_i \sum_j W_{ij}$ be the total volume to be transported. Roughly $\frac{p-1}{p}W$ is transported *between hubs* if each hub is utilized by the same amount of flow, so that we expect $\frac{p-1}{p}\frac{W}{q}$ vehicles to be used on the $p(p-1)$ hub-hub-connections. The expected number of vehicles for each connection is thus

$$e_v = \frac{\frac{p-1}{p}\frac{W}{b}}{p(p-1)} = \frac{W}{p^2 q}$$

We want to choose three reasonable cases, for which we take $e_v = 1$, $e_v = 2$, and $e_v = 4$. This leads to the three vehicles sizes

$$q_1 = \frac{W}{p^2} \qquad q_2 = \frac{W}{2p^2} \qquad q_4 = \frac{W}{4p^2}$$

As these sizes are multiples of each other, each solution for q_1 leads to a solution for q_2 with the same objective (and doubled vehicle numbers) and likewise for q_2 and q_4. This means that the optimal solution for q_4 is at least as good as the one for q_2 or q_1. The datasets that are derived from AP by using q_1, q_2 and q_4 will be called AP1, AP2 and AP3.

5.2. Results

In Table 1 we report the root node gaps and execution times for medium and large-size instances of our test bed. For that, we used the Lagrangian relaxation (without the branch-and-bound algorithm) and compared it to the LP-solver of Gurobi.

The formula we used to compute the percent gaps is $100 \times (\text{Opt} - \text{Lb})/\text{Lb}$, where Lb stands for the value of the lower bound. The first four columns of the table give the problem size (n), the number of desired hubs (p), the type of the instances (type), and the optimal solution to the problem (Optimal). The optimal solutions we reported here are those obtained at the end of the branch-and-bound algorithm. However, for those instances with unknown optimal solution, we give the value of the best feasible solution obtained by the Lagrangian approach; they are labelled by a "†" sign. The next two columns, from left to right, give the relative gaps in percent and the execution times obtained using the Lagrangian approach (Lag). The last two columns of the table present the percent gaps and the execution times of solving the LP relaxation of LP4.

As we can see, the relative gap for the AP1 instances is much better for the Lagrangian approach. For this data set, the average gap of the Lagrangian approach is 1.87%, while the average gap of the LP4 is 7.97%. Moreover, for 10 of the 12 considered instances of the

Table 1: Lower bound comparisons on selected instances.

Instance				Lag		LP4	
n	p	type	Optimal	gap (%)	time (s)	gap (%)	time (s)
25	2	AP1	182302.52	0.00	11.10	3.71	20.51
25	3	AP1	168759.25	2.78	80.64	8.00	20.51
25	4	AP1	152222.68	1.97	73.24	8.57	21.47
25	5	AP1	142974.39	2.76	67.15	13.57	18.53
40	2	AP1	186443.02	0.00	239.98	4.81	597.73
40	3	AP1	171238.33	2.14	484.60	7.25	632.20
40	4	AP1	157342.58	2.35	405.88	8.50	607.52
40	5	AP1	149342.84	3.35	369.04	10.10	634.54
50	2	AP1	186795.36	0.00	1007.32	4.45	1859.81
50	3	AP1	171049.98	1.88	1178.62	7.30	1913.95
50	4	AP1	155877.36	1.68	1020.17	8.02	2204.48
50	5	AP1	149399.31†	3.67	881.11	11.40	2194.44
25	2	AP2	182302.52	3.66	91.76	3.71	19.55
25	3	AP2	159134.60	1.48	74.29	2.44	20.22
25	4	AP2	147353.22	4.60	69.87	5.55	21.75
25	5	AP2	132259.26	5.15	63.03	6.57	18.79
40	2	AP2	186443.02	4.65	544.50	4.81	608.62
40	3	AP2	164353.22	2.68	416.03	3.36	631.27
40	4	AP2	150565.99†	3.80	360.93	4.38	612.14
40	5	AP2	140998.29†	4.01	340.26	4.78	634.08
50	2	AP2	186795.36	4.56	1451.70	4.45	2137.05
50	3	AP2	163518.48	2.61	1016.75	3.03	2177.54
50	4	AP2	149989.81†	4.05	860.39	4.41	2159.06
50	5	AP2	149399.31†	9.59	791.19	11.40	2167.87
25	2	AP3	178956.71	1.90	83.99	1.91	19.67
25	3	AP3	158680.05	2.13	73.15	2.16	20.18
25	4	AP3	142297.49	2.10	68.83	2.19	21.66
25	5	AP3	129157.49	3.79	62.66	4.32	18.98
40	2	AP3	180304.15	1.48	504.93	1.57	606.48
40	3	AP3	161763.22	1.64	412.90	1.81	627.39
40	4	AP3	147864.19	2.56	381.34	2.63	619.88
40	5	AP3	137937.27	2.60	343.64	2.66	641.81
50	2	AP3	180723.19	1.24	1214.51	1.24	2111.19
50	3	AP3	160624.06	1.21	967.38	1.28	2155.72
50	4	AP3	146464.25	2.46	850.96	2.11	2201.40
50	5	AP3	136799.24†	3.17	810.19	3.24	2190.12

† The best-known (not necessarily optimal) values.

Table 2: USApHMPv Branch and Bound results for AP1 instances.

Instance			BB_1		BB_0		LP4	
n	p	Optimal	nodes	time (s)	nodes	time (s)	nodes	time (s)
10	2	173350.14	0	0.24	0	0.16	0	0.67
10	3	144829.78	18	4.49	22	4.48	0	1.68
10	4	124100.83	4	3.90	20	4.12	0	2.38
10	5	114430.41	78	5.41	116	6.28	680	3.90
20	2	181002.24	0	1.99	0	7.77	0	26.35
20	3	163737.01	12	35.40	78	41.51	702	1141.10
20	4	151673.42	52	43.21	198	60.76	7709	14053.29
20	5	143953.12	1094	173.66	3438	385.18	–	–
25	2	182302.52	0	5.40	0	21.26	0	145.70
25	3	168759.25	60	99.59	340	150.62	–	–
25	4	152222.68	14	86.08	70	102.99	–	–
25	5	142974.39	218	191.90	900	352.74	–	–
40	2	186443.02	0	109.78	0	170.70	–	–
40	3	171238.33	130	685.78	1918	1736.81	–	–
40	4	157342.58	182	720.30	1352	1543.56	–	–
40	5	149342.84	1320	3806.78	16338	13541.79	–	–
50	2	186795.36	0	468.04	0	828.27	–	–
50	3	171049.98	142	1729.04	2646	4938.93	–	–
50	4	155877.36	200	1904.38	1964	4774.47	–	–
50	5	149399.31†	11266	29961.42	–	–	–	–

† The best-known (not necessarily optimal) values.

AP1 data set, the gap of the Lagrangian approach is below 3% while this number is zero for LP4. Concerning the running times of the two approaches, the average running time of the Lagrangian approach is about 484 seconds while for LP4 it is about 894 seconds. Thus, the Lagrangian approach outperforms the LP4 in terms of both the gap and the running time. This allowed us to solve all (except one) AP1 instances to optimality (see Table 2) using the Lagrangian approach.

The average gap of the Lagrangian approach for the AP2 and AP3 data sets are closer to those obtained by LP4. For the AP2 and AP3 data sets the average gaps for the Lagrangian approach are 4.23% and 2.19% while these values for the LP4 are 4.90% and 2.26%. As expected, these two data sets are closer to the classical (linear) problem formulation, so that the bound in the root node of LP-based approaches is not so bad any more. However, in terms of the average running time, the Lagrangian approach always outperforms the LP4, which again allowed us to solve most of these instances to optimality within the time limit (see Tables 3 and 4). We should note here that the LP4 approach is not only much slower,

Table 3: USApHMPv Branch and Bound results for AP2 instances.

Instance			BB_1		BB_0		LP4	
n	p	Optimal	nodes	time (s)	nodes	time (s)	nodes	time (s)
10	2	173350.14	8	4.73	20	4.73	0	0.88
10	3	143373.48	38	4.69	84	5.37	84	1.80
10	4	118445.00	36	4.46	78	5.18	137	1.65
10	5	99695.31	52	4.79	80	5.25	59	1.07
20	2	181002.24	54	41.38	154	48.89	79	89.90
20	3	156707.63	30	34.72	148	42.38	359	441.58
20	4	145861.22	440	77.66	1590	168.43	4424	3100.47
20	5	132402.88	582	98.63	2082	230.91	8058	4811.19
25	2	182302.52	72	104.43	286	137.18	26	467.77
25	3	159134.60	8	76.37	48	85.81	144	2571.44
25	4	147353.22	438	173.95	2204	454.26	–	–
25	5	132259.26	738	229.69	2970	623.86	–	–
40	2	186443.02	592	1005.46	3712	2550.51	–	–
40	3	164353.49	94	565.39	2026	1593.77	–	–
40	4	150565.99†	1056	1554.13	–	–	–	–
40	5	140998.29†	7704	7012.02	–	–	–	–
50	2	186795.36	3136	5328.82	25752	26433.72	–	–
50	3	163518.48	132	1636.32	1366	3110.40	–	–
50	4	149989.81†	8414	14026.93	–	–	–	–
50	5	149399.31†	–	–	–	–	–	–

† The best-known (not necessarily optimal) values.

but also uses huge amounts of RAM, up to 50 GB for the largest instances.

In Tables 2, 3, and 4 we present the results for our branch-and-bound algorithm (based upon the Lagrangian relaxation) for the AP1, AP2 and AP3 data sets, respectively. Note that in the following paragraphs "branch-and-bound" algorithm always refers to our branch-and-bound algorithm and not the MIP-Solver of Gurobi, which is used in comparison.

We compare the results of the branch-and-bound algorithm with those of solving LP4 by the Gurobi MIP-Solver. For comparison, we also solved the instances up to a gap of at most 1% to reduce the size of the branch-and-bound tree. We also imposed a time limit of 12 hours to all approaches. In the tables, BB_0 and BB_1 stand for the branch-and-bound algorithm with the gap of 0% and 1%, respectively. In all tables, we report for each algorithm the number of branch-and-bound nodes (nodes) and the overall computing time (time) in seconds (considering only those instances solved to optimality resp. 1% gap). An entry "–" indicates that the algorithm was not able to solve the instance to optimality (or 1% gap) within the time limit.

Table 4: USApHMPv Branch and Bound results for AP3 instances.

Instance			BB_1		BB_0		LP4	
n	p	Optimal	nodes	time (s)	nodes	time (s)	nodes	time (s)
10	2	170386.35	4	4.58	18	4.46	0	0.72
10	3	139703.73	28	4.36	58	4.72	69	1.18
10	4	116539.12	46	4.75	102	5.58	190	1.08
10	5	95821.78	50	4.60	78	5.09	24	0.92
20	2	176913.81	22	35.87	92	41.07	34	45.71
20	3	155838.63	104	40.18	422	63.30	1263	1089.49
20	4	140858.60	104	42.85	422	67.94	1273	1022.59
20	5	128412.19	590	91.54	2132	220.17	8422	2893.22
25	2	178956.71	40	89.71	240	114.91	35	201.35
25	3	158680.05	26	79.37	194	111.37	247	4603.12
25	4	142297.50	72	89.29	430	144.62	626	5220.99
25	5	129157.49	950	241.59	4170	753.94	8376	17542.48
40	2	180304.15	10	504.65	142	580.21	0	39410.02
40	3	161763.22	24	449.95	792	872.90	–	–
40	4	147864.19	438	820.92	10912	6513.30	–	–
40	5	137937.27	996	1382.19	19100	11750.21	–	–
50	2	180723.19	12	1315.61	518	1929.93	–	–
50	3	160624.06	6	993.13	216	1345.32	–	–
50	4	146464.25	242	1757.31	7088	9180.42	–	–
50	5	136799.24†	7868	12294.25	–	–	–	–

† The best-known (not necessarily optimal) values.

As we can observe from these tables, our BB_0 algorithm is able to solve most of the instances to optimality in the time limit of 12 hours. More precisely, the BB_0 is able to solve to optimality 19 out of 20 instances of AP1, 16 out of 20 instances of AP2, and 19 out of 20 instances of AP3. For medium and large-size instances of our test bed ($25 \leq n \leq 50$), the BB_0 can solve 11 out of 12 instances of AP1, 8 out of 12 instances of AP2, and 11 out of 12 instances of AP3, while Gurobi can solve LP4 only for 1 instance of AP1, 2 instances of AP2, and 5 instances of AP3 from the medium and large-size instances. The results of our BB_1 are very interesting. The BB_1 was considerably faster and led to the same upper bound as BB_0 in most of the cases. This means that a large part of the solution effort is only necessary for raising the lower bound for the last percent. It should also be noted that both BB_0 and BB_1 only use a moderate amount of RAM, namely about 2 GB for the largest instances, while as we mentioned earlier, solving LP4 using Gurobi uses huge amounts of RAM, up to 50 GB.

6. Conclusion

In this paper we introduced a new version of the Hub Location Problem in which vehicle sizes are included directly in the model. This leads to the usage of a stepwise cost function, where the step width is the capacity of the vehicle. Introducing vehicle sizes into hub location problems makes them more realistic—but this comes at the cost of a weak LP relaxation. Therefore, we developed a Lagrangian relaxation scheme exploiting the problem structure to obtain tight upper and lower bounds, and proposed a branch-and bound algorithm to solve the problem to optimality. The results of our branch-and bound algorithm are very promising as we could solve instances with up to 50 nodes to optimality.

7. Acknowledgements

The first and second author were supported by the German Research Foundation (DFG) under grants BU 2313/2 and CL 318/14, respectively.

[1] M. E. O'Kelly, A quadratic integer program for the location of interacting hub facilities, European Journal of Operational Research 32 (3) (1987) 393–404.
[2] M. E. O'Kelly, Hub facility location with fixed costs, Papers in Regional Science 71 (3) (1992) 293–306.
[3] A. T. Ernst, M. Krishnamoorthy, Efficient algorithms for the uncapacitated single allocation p-hub median problem, Location science 4 (3) (1996) 139–154.
[4] A. T. Ernst, M. Krishnamoorthy, Solution algorithms for the capacitated single allocation hub location problem, Annals of Operations Research 86 (1999) 141–159.
[5] D. Skorin-Kapov, J. Skorin-Kapov, M. O'Kelly, Tight linear programming relaxations of uncapacitated p-hub median problems, European Journal of Operational Research 94 (3) (1996) 582–593.
[6] A. Ilić, D. Urošević, J. Brimberg, N. Mladenović, A general variable neighborhood search for solving the uncapacitated single allocation p-hub median problem, European Journal of Operational Research 206 (2) (2010) 289–300.
[7] A. Bailey, B. Ornbuki-Berrnan, S. Asobiela, Discrete pso for the uncapacitated single allocation hub location problem, in: Computational Intelligence In Production And Logistics Systems (CIPLS), 2013 IEEE Workshop on, IEEE, 2013, pp. 92–98.
[8] H. Topcuoglu, F. Corut, M. Ermis, G. Yilmaz, Solving the uncapacitated hub location problem using genetic algorithms, Computers & Operations Research 32 (4) (2005) 967–984.
[9] I. Contreras, J. A. Díaz, E. Fernández, Lagrangean relaxation for the capacitated hub location problem with single assignment, OR spectrum 31 (3) (2009) 483–505.
[10] I. Contreras, J.-F. Cordeau, G. Laporte, Benders decomposition for large-scale uncapacitated hub location, Operations research 59 (6) (2011) 1477–1490.
[11] I. Contreras, J. A. Díaz, E. Fernández, Branch and price for large-scale capacitated hub location problems with single assignment, INFORMS Journal on Computing 23 (1) (2011) 41–55.
[12] S. A. Alumur, B. Y. Kara, Network hub location problems: The state of the art, European Journal of Operational Research 190 (1) (2008) 1–21.
[13] J. F. Campbell, M. E. O'Kelly, Twenty-five years of hub location research, Transportation Science 46 (2) (2012) 153–169.
[14] C. B. Cunha, M. R. Silva, A genetic algorithm for the problem of configuring a hub-and-spoke network for a ltl trucking company in brazil, European Journal of Operational Research 179 (3) (2007) 747–758.
[15] R. S. de Camargo, G. de Miranda Jr, H. P. L. Luna, Benders decomposition for hub location problems with economies of scale, Transportation Science 43 (1) (2009) 86–97.
[16] G. Song, Integer programming models for airline network design problems, Ph.D. thesis, University of Texas at Austin (1996).

[17] R. van de Leensel, Models and algorithms for telecommunication network design, Ph.D. thesis, Maastricht University (1999).
[18] J. Sender, U. Clausen, Hub location problems with choice of different hub capacities and vehicle types, in: Network Optimization, Springer, 2011, pp. 535–546.
[19] J. F. Meier, U. Clausen, Some numerical studies for a complicated hub location problem, in: H.-J. Sebastian, P. Kaminsky, T. Müller (Eds.), Quantitative Approaches in Logistics and Supply Chain Management, Springer, 2014, pp. 33–43.
[20] J. F. Meier, U. Clausen, Strategic planning in ltl logistics – increasing the capacity utilization of trucks, Electronic Notes in Discrete Mathematics 41 (2013) 37–44.
[21] B. Fleischmann, Transport- und Tourenplanung, in: D. Arnold, H. Isermann, A. Kuhn, H. Tempelmeier, K. Furmans (Eds.), Handbuch Logistik, Vol. 3, Springer, 2008, pp. 137–152.
[22] K. Haase, M. Hoppe, Transportnetzgestaltung für Paketdienstleister, Zeitschrift für Betriebswirtschaft 78 (9) (2008) 857–874.
[23] M. N. Baumung, H. I. Gündüz, Consolidation of residual volumes in a parcel service provider's long-haul transportation network, in: Computational Logistics, Springer, 2015, pp. 437–450.
[24] J. F. Meier, U. Clausen, B. Rostami, C. Buchheim, A compact linearisation of Euclidean single allocation hub location problems, Electronic Notes in Discrete Mathematics Accepted for Publication.
[25] J. F. Meier, U. Clausen, Solving classical and new single allocation hub location problems on euclidean data, Tech. rep., Optimization Online, http://www.optimization-online.org/DB_HTML/2015/03/4816.html (2015).
[26] H. Yaman, G. Carello, Solving the hub location problem with modular link capacities, Computers & Operations Research 32 (12) (2005) 3227 – 3245. doi:http://dx.doi.org/10.1016/j.cor.2004.05.009.
URL http://www.sciencedirect.com/science/article/pii/S0305054804001133
[27] G. B. Dantzig, Discrete-variable extremum problems, Operations Research 5 (2) (1957) 266–288. arXiv:http://dx.doi.org/10.1287/opre.5.2.266, doi:10.1287/opre.5.2.266.
URL http://dx.doi.org/10.1287/opre.5.2.266
[28] B. Korte, J. Vygen, Combinatorial Optimization: Theory and Algorithms, Springer, 2012, Ch. 17.1 Fractional Knapsack and Weighted Median.
[29] M. Held, R. M. Karp, The traveling-salesman problem and minimum spanning trees: Part ii, Mathematical Programming 1 (1) (1971) 6–25. doi:10.1007/BF01584070.
URL http://dx.doi.org/10.1007/BF01584070
[30] J. E. Beasley, OR library, http://people.brunel.ac.uk/~mastjjb/jeb/orlib/phubinfo.html (2012).