An Efficient and Scalable Approach to Hub Location Problems based on Contraction

Sebastian Wandelt[a,b], Weibin Dai[a], Jun Zhang[a], Qiuhong Zhao[c], Xiaoqian Sun[a,*]

[a]National Key Laboratory of CNS/ATM, School of Electronic and Information Engineering, Beihang University, 100191 Beijing, China
[b]Institute of Computer Science, Humboldt University of Berlin, 12489 Berlin, Germany
[c]School of Economics and Management, Beihang University, 100191, Beijing, China

Abstract

Solving the vast majority of hub location problems is NP-hard, implying that optimally solving large-scale instances (with hundreds of nodes) with exact solution techniques is extremely difficult. While heuristics have been developed which scale up to hundreds of nodes for specific problem types, these techniques do not scale up for further larger instances (with thousands of nodes) or intriguing problem variants.

In this paper, we propose EHLC (Efficient Hub Location by Contraction), which exploits the idea of efficiently computing hub locations on a reduced network instance, so-called contracted network. The obtained solutions are rewritten back to the original network, followed by a final optimization step. A rich set of computational experiments on instances with up to 5000 nodes and different problem types, i.e., USApHMPC, CSApHMPC, USApHMPI, UMApHMPC, CMApHMPC, and UMApHMPI shows that EHLC outperforms the existing solution techniques by orders of magnitude regarding execution time, while achieving solutions with identical gaps for almost all datasets and parameter combinations. For large enough datasets or complex hub location problems, EHLC has a speedup of over 20 times (such as GA, GVNS for USApHMPI on URAND1000 and Benders for UMApHMPI on TR40), compared to non-contracted methods. Given the same time limit, EHLC provides final solutions with similar or better qualities for most instances, such as EHLC_GVNS and NC_GVNS reach the optimal solutions for most instances.

Keywords: Location, Hubs, Contraction, Scalability

1. Introduction

Hub location problems optimize the location of hubs in a network (O'Kelly 1987), with important applications in transportation (Gelareh and Nickel 2011, O'Kelly 2012) and telecommunication systems (Yaman and Carello 2005,

*Corresponding author: Tel.: + 86 10 8233 8036
Email addresses: wandelt@informatik.hu-berlin.de (Sebastian Wandelt), daiweibin@buaa.edu.cn (Weibin Dai), buaazhangjun@vip.sina.com (Jun Zhang), qhzhao@buaa.edu.cn (Qiuhong Zhao), sunxq@buaa.edu.cn (Xiaoqian Sun)

Kim and O'Kelly 2009). Demands between origin nodes and destination nodes are collected, transferred and distributed by these hub facilities. Economies of scale provide cost discounts for the transportation between hubs. Since the seminal work by O'Kelly (1986), hub location problems have been studied intensively for more than three decades. A large number of exact solution techniques has been proposed in the literature, e.g., based on Lagrangian relaxation, Benders decomposition, branch-and-price, and branch-and-cut. Apart from a few exceptions, hub location problems are NP-hard (Yang et al. 2013, Sadeghi et al. 2015). Given the inherent difficulty to solve large-scale instances towards optimality, a wide range of meta-heuristics have been proposed, which, essentially, aim to enumerate feasible solutions and identify a solution with minimum cost, such heuristics include Tabu search, variable neighborhood search, and genetic algorithms. Depending on the problem types and sizes of the networks, these heuristics require minutes to hours of computation time.

It is well known that the quality of a hub assignment largely depends on the distribution of travel demands between nodes and their spatial positions in the network (O'Kelly 1992, Peker et al. 2015). Several meta-heuristics exploit this rule-of-thumb when creating initial solutions or limiting the search space, by preferably selecting hubs with high demand or central positions in the network. Beyond this rather implicit use, spatial insight is not exploited further in the existing literature. Based on the above discussion, the following question arises: Can we locate the hub facilities and spoke assignments for a given network by solving a *reduced-size* instance first and exploit insights gained from its solutions? Such a novel view and design has the potential to lead towards tremendously reduced computation times, while partially preserving and utilizing the spatial structure present in the original network.

In this study, we propose EHLC (Efficient Hub Location by Contraction). EHLC transforms the input network, for which we are seeking an assignment, into a smaller network with similar topological and demand properties (see Section 3 for details). In order to evaluate the performance of EHLC, five datasets (TR, AP, URAND1000, WORLDAP, and RAND5000) are used as case studies. The general variable neighborhood search (GVNS), genetic algorithms (GA), and Benders decomposition are compared and applied to solve the contracted/original problems. Our experiments on real-world datasets reveal that the solutions obtained by EHLC are highly competitive; state-of-the-art algorithms/heuristics need orders of magnitude more computation time to identify similar solutions for large-scale problems. While we evaluate EHLC on several standard types of hub location problems, we believe that the methodology can be applied to a much wider range of other hub location problems. We hope that EHLC will become a prolific framework for all kinds of large-scale hub location problems.

Contraction for USAPHMPC has been presented at the 2017 IEEE Symposium Series on Computational Intelligence (Dai et al. 2017); this work is extended as follows. First, we lift our contraction methodology from heuristics to exact methods. A new strategy for the contraction step has been designed. The network is contracted by merging nodes with close distance and similar distribution of travel demands. Second, the original problems are resolved by using the rewritten solutions as the initial inputs. It further improves the qualities of final solutions. In addition, in the new strategy, the nodes that are not selected as representatives can sometimes appear as hubs in the final solution. Third, we have used multiple methods (GA, GVNS, Benders decomposition) to solve the contracted problem and resolve the original problem. In addition, multiple types of hub location problems (USApHMPC, USApHMPI, CSApHMPC, UMApHMPC, CMApHMPC, and UMApHMPI) have been solved by EHLC. Compared to USApHMPC in Dai et al. (2017), other types hub location problems have different challenges to solve. For instance, USApHMPI and UMApHMPI need to determine the connection of hub links; CSApHMPC and CMApHMPC need to consider the capacities of hub nodes. These results emphasize the generality of EHLC, for not only different solution techniques, but also different types of hub location problems. Finally, we have extended the range of experiments significantly, by performing sensitivity analysis on the contraction size and reporting experiments with networks of size up to 5,000 nodes. In the real world, there are over 50,000 airports/airfields; about 3,300 of them have scheduled services. Accordingly, there is a need for techniques with more than 1,000 nodes. In the literature, although some algorithms provide good solutions for a specific hub location problem in large-scale networks (such as GVNS for USApHMPC with 1,000 nodes), the adaption of these algorithms to other problems is often very hard or has poor performance. The key contribution of EHLC is its generality. EHLC, which reduces the runtime and keeps the solution quality for different types of hub location problems, is very competitive in that regard. We have further emphasized this motivation in the introduction.

The remainder of this paper is organized as follows. We review the literature on hub location problems in Section 2. The formulations of six types of hub location problems are also provided in this section. The rationale and process of EHLC are proposed in Section 3. To evaluate the performance of EHLC, experiments on the TR, AP, URAND1000, WORLDAP and RAND5000 datasets as case studies are presented in Section 4. The paper concludes with Section 5.

2. Literature review

Hub location problems were introduced by O'Kelly (1986), together with the first mathematical formulation for the p-hub median problem (pHMP) (O'Kelly 1987). Since then, the class of fundamental hub location problems usually contains (single/multiple allocation) p-hub median problems (Campbell 1996, 2009), uncapacitated hub location problems, p-hub center problems, and hub covering problems (Campbell and O'Kelly 2012). Additional constraints lead to many variants, such as capacitated p-HLP (Hoff et al. 2017), hub-arc location problems (Campbell et al. 2003, 2005), continuous p-HLP (Campbell 1993), profit maximizing hub location problems (Taherkhani and Alumur 2019), and multi-objective p-HLP. We refer the readers to the following reviews for further details (Campbell and O'Kelly

2012, Farahani et al. 2013). Here, we only review the model formulations of six types of hub location problems which are the focus of our study. Following the nomenclature in the literature (Campbell and O'Kelly 2012, Farahani et al. 2013), we call them USApHMPC (Uncapacitated Single Allocation p-Hub Median Problem with Complete hub network), CSApHMPC (Capacitated Single Allocation p-Hub Median Problem with Complete hub network), USApHMPI (Uncapacitated Single Allocation p-Hub Median Problem with Incomplete hub network), UMApHMPC (Uncapacitated Multiple Allocation p-Hub Median Problem with Complete hub network), CMApHMPC (Capacitated Multiple Allocation p-Hub Median Problem with Complete hub network), and UMApHMPI (Uncapacitated Multiple Allocation p-Hub Median Problem with Incomplete hub network), respectively. For each HLP, we describe the formulation and refer readers to the related work for further information.

1. **USApHMPC: Uncapacitated single allocation p-hub median problem with complete hub network**

   In USApHMPC, each pair of hubs is connected with a link, yielding a complete hub network, and each node is allocated to a single hub only. Let $G = (V, E)$ be a network, where V and E are the set of nodes and links between nodes, respectively. The number of nodes is n and the number of hubs is p. For each pair of nodes *(i,j)*, let c_{ij} and w_{ij} be the cost and travel demand between them. Let $O_i = \sum_{j \in V} w_{ij}$ be the total travel demand from the source node i and $D_i = \sum_{j \in V} w_{ji}$ be the total travel demand to the destination node i (Contreras et al. 2009b). USApHMPC is formulated as follows (Ernst and Krishnamoorthy 1996):

$$\min \sum_{i \in V} \sum_{k \in V} c_{ik} Y_{ik} (\delta_1 O_i + \delta_2 D_i) + \sum_{i \in V} \sum_{k \in V} \sum_{m \in V} \alpha c_{km} X^i_{km} \tag{1}$$

$$\text{subject to} \sum_{k \in V} Y_{ik} = 1, \forall i \in V \tag{2}$$

$$\sum_{k \in V} Y_{kk} = p \tag{3}$$

$$Y_{ik} \le Y_{kk}, \forall i, k \in V \tag{4}$$

$$\sum_{m \in V, m \neq k} X^i_{km} - \sum_{m \in V, m \neq k} X^i_{mk} = O_i Y_{ik} - \sum_{j \in V} w_{ij} Y_{jk}, \forall i, k \in V \tag{5}$$

$$Y_{ik} \in \{0, 1\}, \forall i, k \in V \tag{6}$$

$$X^i_{km} \ge 0, \forall i, k, m \in V \tag{7}$$

   Here, the objective function (1) is the sum of transportation costs. Parameters $\alpha < 1, \delta_1 > \alpha, \delta_2 > \alpha$ are the cost coefficients for transporting travel demands between hub nodes, from spoke nodes to hub nodes, and from hub nodes to spoke nodes, respectively. Variables X^i_{km} represent the flows routed on hub link *(k,m)* originating from node i. Let Y_{ik} be the allocation variable defined below (Ernst and Krishnamoorthy 1996):

$$
Y_{ik} = \begin{cases} 1, & \text{if node } i \text{ is assigned to hub } k \\ 0, & \text{otherwise} \end{cases}
$$

Equation (2) and Equation (4) ensure that each node is assigned to one hub only and it can only be assigned to a hub. Equation (3) fixes the number of hubs to p. Equation (5) ensures the flow equilibrium from each node i for each hub k.

Since USApHMPC is a well-known variant of hub location problems, many algorithms have been proposed, including Benders decomposition (Ghaffarinasab and Kara 2019), Lagrangian relaxation (Pirkul and Schilling 1998), general variable neighborhood search (Ilić et al. 2010), genetic algorithms (Kratica et al. 2007), and Tabu search (Skorin-Kapov and Skorin-Kapov 1994).

2. **CSApHMPC: Capacitated single allocation p-hub median problem with complete hub network**

In transportation systems, hubs usually have limited capacities. For instance, Hartsfield-Jackson Atlanta International Airport is frequently operating at its maximum throughput, defined by terminal capacity or number of arrivals/departures. Accordingly, in CSApHMPC, a new constraint enforces that the total flow through each hub cannot exceed a specific capacity. Using parameter λ_i to represent the capacity of node i, in addition to Equations (1–7), the following constraint is required for CSApHMPC (Ernst and Krishnamoorthy 1999):

$$
\sum_{i \in V} O_i Y_{ik} \leq \lambda_k Y_{kk}, \forall k \in V \tag{8}
$$

Note that a more generic version of capacity constraint, which computes the total travel flow going through the hub node, was proposed by Campbell (1994). However, only one capacitated HLP (CSApHMPC) is studied in this paper. Equation (8) with lower complexity is enough for this problem. Because of the capacity constraint, it is more difficult to solve CSApHMPC compared to USApHMPC. Several methods, such as Benders decomposition (Rodríguez-Martín and Salazar-González 2008), Lagrangian relaxation (Contreras et al. 2009a), and genetic algorithms (Stanimirović 2012), have been proposed to solve CSApHMPC and similar hub location problems.

3. **USApHMPI: Uncapacitated single allocation p-hub median problem with incomplete hub network**

While USApHMPC assumes that the hub network is complete, this is not necessarily the case in real transportation systems. In USApHMPI, a subset of links between hubs are present only, leading to an incomplete hub network. Accordingly, in addition to the selection of hubs and allocation of spoke nodes, links between hubs also need to be determined. Let $Z_{km} (\forall k, m < k \in V)$ be the decision variable:

$$Z_{km} = \begin{cases} 1, & \text{if hub } k \text{ and hub } m \text{ are connected} \\ 0, & \text{otherwise} \end{cases}$$

The formulation of USApHMPI with q hub links is shown as below (Alumur et al. 2009):

$$\min \sum_{i \in V} \sum_{k \in V} c_{ik} Y_{ik} (\delta_1 O_i + \delta_2 D_i) + \alpha \sum_{i \in V} \sum_{k \in V} \sum_{m \in V} c_{km} X^i_{km} \tag{9}$$

$$\text{subject to} \sum_{k \in V} Y_{ik} = 1, \forall i \in V \tag{10}$$

$$\sum_{k \in V} Y_{kk} = p \tag{11}$$

$$\sum_{k \in V} \sum_{m \in V, m < k} Z_{km} = q \tag{12}$$

$$Y_{ik} \leq Y_{kk}, \forall i, k \in V \tag{13}$$

$$Z_{km} \leq Y_{kk}, \forall k, m < k \in V \tag{14}$$

$$Z_{km} \leq Y_{mm}, \forall k, m < k \in V \tag{15}$$

$$\sum_{m \in V, m \neq k} X^i_{km} - \sum_{m \in V, m \neq k} X^i_{mk} = O_i Y_{ik} - \sum_{j \in V} w_{ij} Y_{jk}, \forall i, k \in V \tag{16}$$

$$X^i_{km} + X^i_{mk} \leq O_i Z_{km}, \forall i, k, m < k \in V \tag{17}$$

$$X^i_{km} \geq 0, \forall i, k, m \neq k \in V \tag{18}$$

$$Y_k \in \{0, 1\}, \forall k \in V \tag{19}$$

$$Z_{km} \in \{0, 1\}, \forall k, m < k \in V \tag{20}$$

Equation (12) forces that the number of links between hubs is q. Equations (14–15) ensure that links can be established between hubs only. Equations (16–17) are the flow equilibrium constraint and flow capacity constraint, respectively. In the literature, few algorithms have been proposed for solving USApHMPI specifically, mixed integer programming is used in most cases (Alumur et al. 2009). Extensions of this problem with additional constraints can be solved with Tabu search (Calık et al. 2009).

4. **UMApHMPC: Uncapacitated multiple allocation p-hub median problem with complete hub network**

USApHMPC assumes that each spoke is connected to exactly one hub. In real transportation systems, however, passengers/cargo can be routed through different hubs, depending on the destination, which leads to further reduced costs. Consequently, in UMApHMPC, each spoke node is assigned to more than one hub. Although a formulation with $O(n^3)$ variables has been proposed by Ernst and Krishnamoorthy (1998), it is difficult to be solved with several general solution techniques, such as Benders decomposition (de Camargo et al. 2009). Therefore, researchers often use a formulation with $O(n^4)$ variables which was initially proposed by Campbell (1994). Assume variable Y_k is used to represent the location of hubs:

$$Y_k = \begin{cases} 1, & \text{if node } k \text{ is a hub} \\ 0, & \text{otherwise} \end{cases}$$

The path for the demand from node i to node j is represented by variable X_{ijkm}:

$$X_{ijkm} = \begin{cases} 1, & \text{if the demand from node } i \text{ to node } j \text{ is routed through hubs } k \text{ and } m \\ 0, & \text{otherwise} \end{cases}$$

Overall, UMApHMPC is formulated as follows:

$$\min \sum_{i \in V} \sum_{j \in V} \sum_{k \in V} \sum_{m \in V} \left(\delta_1 c_{ik} + \alpha c_{km} + \delta_2 c_{mj} \right) w_{ij} X_{ijkm} \tag{21}$$

$$\text{subject to } \sum_{k \in V} Y_k = p \tag{22}$$

$$\sum_{k \in V} \sum_{m \in V} X_{ijkm} = 1, \forall i, j \in V \tag{23}$$

$$\sum_{m \in V} X_{ijkm} + \sum_{m \in V, m \neq k} X_{ijmk} \leq Y_k, \forall i, j, k \in V \tag{24}$$

$$Y_k \in \{0, 1\}, \forall k \in V \tag{25}$$

$$X_{ijkm} \in \{0, 1\}, \forall i, j, k, m \in V \tag{26}$$

Here, the objective function (21) is the sum of transportation costs for all OD pairs. Equation (22) ensures that the total number of hubs is equal to p. Equation (23) forces that there is one and only one path between node pair (i, j). Due to Equation (24), only hubs can be used to transport travel demands between nodes. To solve UMApHMPC and similar hub location problems, algorithms such as Benders decomposition (de Camargo et al. 2008, 2009), Lagrangian relaxation (An et al. 2015), and genetic algorithms (Kratica et al. 2005) have been proposed. In this study, we want to solve multiple allocation problems with an efficient exact solution technique. Benders decomposition is the best choice.

5. **CMApHMPC: Capacitated multiple allocation p-hub median problem with complete hub network**

Which is similar with CSApHMPC, a capacity constraint is also added to CMApHMPC. Using parameter λ_i to represent the capacity of node i, in addition to Equations (21–26), the following constraint is required for CMApHMPC (Ebery et al. 2000):

$$\sum_{i \in V} \sum_{j \in V} \sum_{m \in V} X_{ijkm} w_{ij} \leq \lambda_k Y_k, \forall k \in V \tag{27}$$

We did not find the appropriate references about Benders decomposition for CMApHMPC. Therefore, we design this algorithm by ourselves, inspired by de Camargo et al. (2008).

6. **UMApHMPI: Uncapacitated multiple allocation p-hub median problem with incomplete hub network**

Which is similar with USApHMPI, the hub links between hubs also need to be determined for UMAPHMPI. Using binary variables h_{ijk} (and t_{ijm}) to represent whether OD pair (i, j) uses hub k (and hub m) as the first (and the last) hub node. The formulation of UMApHMPI with q hub links is shown as below, based on a simplified version of the model in de Camargo et al. (2017):

$$\min \sum_{i \in V} \sum_{j \in V} w_{ij} \left[\sum_{k \in V} \delta_1 c_{ik} h_{ijk} + \sum_{m \in V} \delta_2 c_{mj} t_{ijm} + \sum_{k \in V} \sum_{m \in V} \alpha c_{km} X_{ijkm} \right] \tag{28}$$

$$\text{subject to } \sum_{k \in V} Y_k = p \tag{29}$$

$$\sum_{m \in V, m \neq j} t_{ijm} + h_{ijj} + \sum_{k \in V, k \neq j} X_{ijkj} = 1, \forall i, j \in V \tag{30}$$

$$h_{ijm} + \sum_{k \in V, k \neq j, k \neq m} X_{ijkm} = \sum_{k \in V, k \neq i, k \neq m} X_{ijmk} + t_{ijm}, \forall i, j, m \in V, i \neq m, j \neq m \tag{31}$$

$$t_{iji} + \sum_{m \in V, m \neq i} X_{ijim} = Y_i, \forall i, j \in V \tag{32}$$

$$h_{ijk} + \sum_{m \in V, m \neq j, m \neq k} X_{ijmk} \leq Y_k, \forall i, j, k \in V, k \neq i, k \neq j \tag{33}$$

$$h_{ijj} + \sum_{k \in V, k \neq j} X_{ijkj} = Y_j, \forall i, j \in V \tag{34}$$

$$\sum_{k \in V} \sum_{m \in V, m < k} Z_{km} = q \tag{35}$$

$$Z_{km} \leq Y_k, \forall k, m < k \in V \tag{36}$$

$$Z_{km} \leq Y_m, \forall k, m < k \in V \tag{37}$$

$$X_{ijkm} \leq (Z_{km} \text{ if } k > m) + (Z_{mk} \text{ if } k < m), \forall i, j, k, m \in V, k \neq j, m \neq i, m \neq k \tag{38}$$

$$Z_{km} \in \{0,1\}, \forall k, m \in V, m < k \tag{39}$$

$$Y_k \in \{0,1\}, \forall k \in V \tag{40}$$

$$X_{ijkm} \in \{0,1\}, \forall i, j, k, m \in V \tag{41}$$

$$h_{ijk} \in \{0,1\}, \forall i, j, k \in V \tag{42}$$

$$t_{ijk} \in \{0,1\}, \forall i, j, k \in V \tag{43}$$

Equation (30) forces that each OD pair (i, j) is served by a path. Equation (31) is the flow equilibrium constraint on each hub m for each OD pair (i, j). Equations (32–34) ensure that the paths between OD pairs can only go through hub nodes. Equation (35) forces that the number of links between hubs is q. Equations (36–37) ensure that links can be established between hubs only. Equation (38) ensures that only hub links can be used for transit. In the literature, Benders decomposition has been used to solve another type of UMApHMPI de Camargo et al. (2017):.

Afterwards, we focus on the description of the solution techniques presented in the literature. In addition to HLPs, the concept of contraction is inspired by the aggregation of demand points in networks. However, a big open question in this topic is how to measure the errors caused by the aggregation. Therefore, we also focus on the error measurements of demand point aggregation in the literature here.

Exact solution techniques: Many hub location problems are formalized as mixed integer programs (MIP). Accordingly, they can be solved with standard MIP solvers, e.g., CPLEX and Gurobi. The runtime and usability of these standard solvers largely depend on the number of variables and constraints in the formulation, the tightness of LP bounds and additional (usually, constant-time) factors. For instance, while the classical formulation of the single allocation p-hub median problem had $O(n^4)$ variables, Ernst and Krishnamoorthy (1996) reduced the number of variables to $O(n^3)$. Such reductions at an order of magnitude often speed-up solution techniques significantly. Albeit this improvement, several other optimization techniques have been introduced in the literature, such as Benders decomposition, Lagrangian relaxation, branch-and-price, and branch-and-cut, as explained below. In Benders decomposition, the original problem is decomposed into a master problem and a sub-problem by keeping the values of some variables fixed (de Camargo et al. 2008, Contreras et al. 2011a, de Camargo et al. 2017, de Sá et al. 2018, Ghaffarinasab and Kara 2019). By solving the dual problem of the sub-problem, one or more new constraints (called Benders cuts) are added to the master problem in each iteration. The upper bound and the lower bound of the problem are updated accordingly. The algorithm terminates when the upper bound and the lower bound converge to the same value. In Lagrangian relaxation, a relaxed problem is generated by relaxing some constraints and adding them to the objective function with several coefficients (called Lagrangian multipliers) (Pirkul and Schilling 1998, An et al. 2015, Contreras et al. 2009a). The lower bound is obtained by solving the relaxed problem. The upper bound is obtained by constructing feasible solutions based on the lower bound. By updating the values of Lagrangian multipliers, the lower bound and the upper bound converge until termination of the algorithm. In branch-and-cut algorithms, a branch-and-bound method and a cutting plane method are used to explore the decision tree and compute the bounds, respectively (Labbé et al. 2005, Rodríguez-Martín and Salazar-González 2008, Rodríguez-Martín et al. 2014). In the search tree, a linear relaxation of the original problem is constructed and the cutting plane method is used to solve the relaxed problem. The branch-and-bound procedure continues afterwards. Branch-and-price, on the other hand, is a combination of column generation and branch-and-bound (Thomadsen and Larsen 2007, Contreras et al. 2011b). The original problem is decomposed into a restricted master problem and a pricing problem. By solving the latter, new columns are generated and added to the former. If no columns can be found and the solution for the relaxed problem is not integer, the branch-and-bound is applied. The techniques above have in common that they derive a lower bound and an upper bound for the problem at hand; once both bounds converge within a predefined threshold, the feasible solution inducing the upper bound is considered as optimal. Finally, Meier and Clausen (2018) proposed a novel method for solving single allocation hub location problems in the Euclidean data. They transforms the quadratic formulations for single allocation problems into linear formulations with $O(n^2)$ variables. A row generation procedure is applied to improve the convergence speed of the algorithm.

Meta-heuristics: Given that the computation of exact solutions is inherently difficult for large problems instances, as most HLPs are NP-hard, a class of solution techniques formally explores (a subset of) feasible solutions, returning the solution with the minimum costs. The exploration of feasible solution space is driven by a search heuristic. Depending on the problem type and search heuristic, solutions with acceptable gaps (deviation of costs from the optimal solution cost) can be obtained. Genetic algorithms (GA) (Azizi et al. 2016), general variable neighborhood search (GVNS) (Ilić et al. 2010), and Tabu search (TS) (Abyazi-Sani and Ghanbari 2016, Karimi 2018) are some commonly used heuristics. In genetic algorithms, each solution is encoded to a chromosome (Kratica et al. 2005, 2007, Stanimirović 2012). An initial population of solutions are generated at the beginning of the algorithm. In each iteration, several solutions are selected as parents. New solutions (called offspring) are generated by applying selection, crossover and, mutation operators. By generating new offspring and discarding poor solutions, genetic algorithms often provide good solutions after sufficient number of iterations. General variable neighborhood search has two phases, i.e., the descent phase and the perturbation phase (Ilić et al. 2010, Todosijević et al. 2017, Brimberg et al. 2017, Dai et al. 2019). In the descent phase, the algorithm finds the local minimum by searching and changing the neighborhood of the current solutions. In the perturbation phase, the algorithm gets out of the solution valley by a random shake operation. General variable neighborhood search performs well on single allocation HLPs because of the specific structure of the problems. Tabu search is another type of search method for solving HLPs (Skorin-Kapov and Skorin-Kapov 1994, Silva and Cunha 2009, Abyazi-Sani and Ghanbari 2016). By using one or more tabu lists, the algorithm avoids searching the same solution (hub location or spoke allocation) repeatedly unless a better solution is obtained.

Domain-specific heuristics: In recent years, some novel methods which consider the spatial properties of networks have also been proposed. Figueiredo et al. (2014) proposed a two-stage method for HLPs. By solving a p-median problem in the first stage, p regional hubs are obtained. Based on these p regional hubs, a q-hub location problem is solved in the original network. Peker et al. (2015) proposed a clustering-based method, which is based on the spatial properties and travel demands between nodes: A potential hub set is generated, which can help to reduce the computational complexity of the problems by narrowing the solution space.

Error measures of contraction/aggregation: The concept of contraction is similar to the aggregation of demand points in some traditional location problems (Plastria 2001). This topic has been well studied by a number of researchers (Francis et al. 2004, Emir-Farinas and Francis 2005). For a general review of the demand point aggregation, just see Rogers et al. (1991), Francis et al. (2002). Although aggregation decreases the cost for designing the model, solving the problem and the uncertainty of the data, it increases the error of the model. It is still an open question that how to make a trade-off between the benefits and the model error (Francis et al. 2009). The aggregation method is usually applied to non-complex location problems. The bounds of both original problem and the aggregated problem are obtained by solving a restriction and a relaxation of the aggregated problem. If the bounds are close to each other, the optimal solution of the aggregated problem can be seen as (nearly) optimal to the original problem. However, in hub location problems, the bounds usually become loose. Therefore, our contraction method is required in this case.

Regarding the above literature review, exact algorithms cannot scale up to large-scale instances (with hundreds of nodes) and heuristics do not scale up for further larger instances (with thousands of nodes) or intriguing problem variants. Although some algorithms may provide good results for a specific problem, the adaption to other problems is often tedious and requires a significant amount of work. Slightly exaggerated, one could say that each pair of problem type and solution technique is often studied in an individual piece of publication.

3. EHLC: A novel method for solving hub location problems by network contraction

3.1. Rationale for EHLC on a running example

Before we describe technical implementation details of EHLC, its general idea is introduced first. The methodology consists of four distinct steps, as shown in Figure 1, which are described below. A corresponding running example

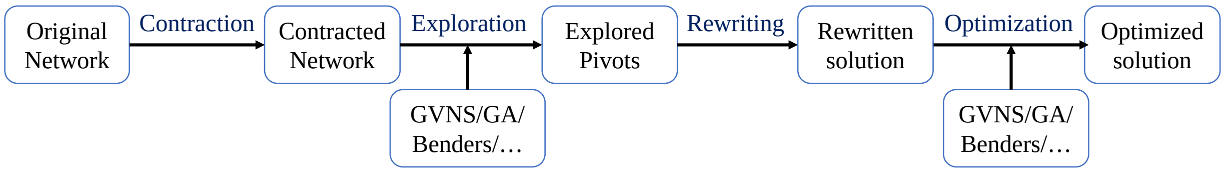


Figure 1: The general process of EHLC as a flowchart, consisting of four individual steps indicated by horizontal arrows.

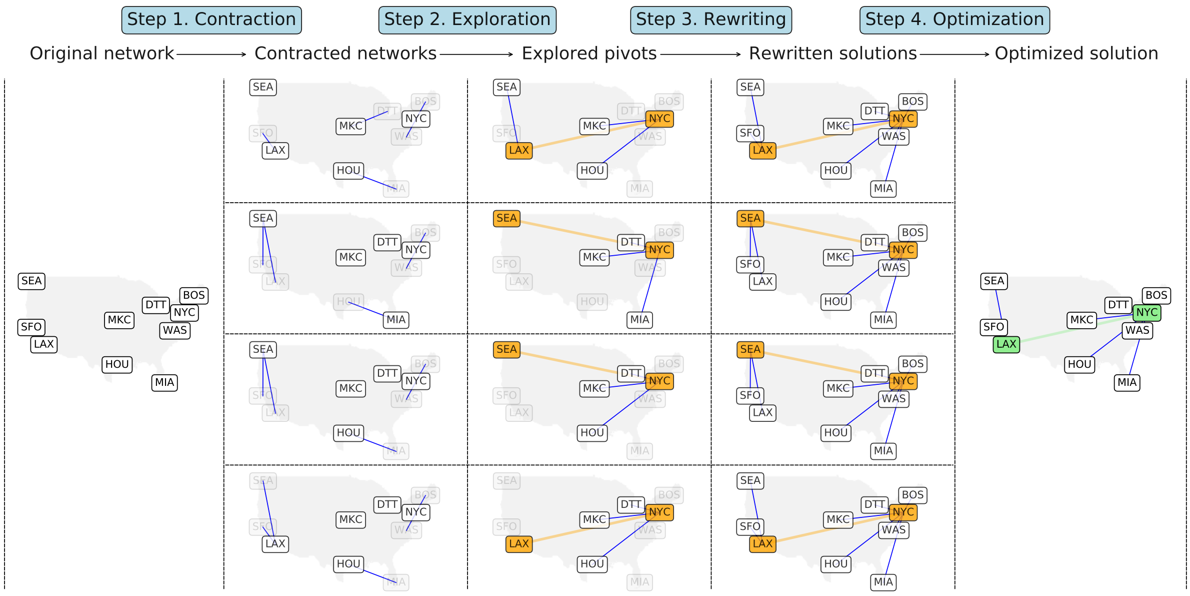


Figure 2: The process of EHLC for a USApHMPC instance on the CAB dataset with ten nodes: The original network with ten nodes is contracted into a smaller network with only five representative nodes. USApHMPC is solved for the contracted network and the contracted solution is obtained. The results are rewritten back to the original network and the final solution is obtained by resolving the original problem based on the rewritten solution. A summarized flowchart is added below the example.

shown in Figure 2:

Step 1. Contraction: Given an original network with n nodes (See *Original network* in Figure 2) and a given number k, we define a contraction function $f: V \to V$ on the set of nodes V such that $|f(V)| = k$, where the set of nodes in the original network is represented by V. Each node $i \in V$ is mapped to a node $s \in V$. Let V^* be the image of function f, i.e., $V^* = f(V)$. The elements in set V^* are called contraction nodes. As shown below the label *Contracted networks* in Figure 2, ten nodes in the original network are mapped to five contraction nodes. Then, a contracted network is constructed with these contraction nodes. For a contraction node s, its travel demand in the contracted network is the sum over all demands of nodes that are mapped to node s. Thus, the demand from contraction node r to contraction node s after contraction is computed by:

$$w^*_{rs} = \sum_{j \in f^{-1}(s)} w_{rj} + \sum_{i \in f^{-1}(r)} w_{is} - w_{rs} = \sum_{i \in f^{-1}(r), j \in f^{-1}(s)} w_{ij}, \forall r, s \in V^* \tag{44}$$

where $f^{-1}(s) = \{j \in V : f(j) = s\}$ is the inverse of f. We define the cost for any pair of contraction nodes *(r,s)*:

$$c^*_{rs} = c_{rs}, \ \forall r, s \in V^*, r \neq s \tag{45}$$

The contracted HLPs are formulated by simply replacing *c,w* and V with c^*, w^* and V^* in Equations (1–26). Accordingly the contracted formulation has the same structure with the original formulation. In total, Figure 2 shows four different contractions, with five nodes each.

Step 2. Exploration: The HLPs in the contracted network can be solved using any method applicable to the original network, since the type of HLP is the same. Depending on the solution technique, one or more feasible solutions with the hub set, spoke assignment, and hub links (in case of USApHMPI) for the contracted network are obtained. As shown below the label *Explored pivots* in Figure 2, the best hub locations (orange nodes), hub links (orange links), and spoke allocations (blue links) are obtained for each contracted network.

Step 3. Rewriting: Pivots from the contracted network induce solutions to the original network by reassigning the nodes in $V \backslash V^*$. Using USApHMPC as an example, let $Assignment$ represent the solution in the contracted network. If node i is allocated to hub k, then $Assignment[i] = k$. We need to reassign nodes in $V \backslash V^*$ to the hubs in the contracted solution. For instance, as shown below the label *Rewritten solutions* in Figure 2, the remaining nodes are assigned to their closest hubs. The solutions for the original problem are obtained afterwards.

Step 4. Optimization: The rewritten solutions are feasible but not necessarily near-optimal. In order to further improve the quality of solutions, we solve the original problem with the existing method in Step 2, using the rewritten solutions from the contracted network as the initial solutions. The quality of solutions and the required runtime significantly depend on the initial input in many solution techniques, such as general variable neighborhood search (Dai et al. 2019). Starting from a *good* initial basis, solution techniques are more likely to terminate earlier and to provide better solutions. The solution after resolving the original problem in the example instance is shown below the label *Optimized solution* in Figure 2.

With a description of the running example in mind, we present the details of EHLC in the following subsections: Contraction (Section 3.2), exploration (Section 3.3), rewriting (Section 3.4), and optimization (Section 3.5).

Table 1: Part of travel demands and costs between nodes in the CAB10 network.

	BOS	DTT	HOU	MKC	LAX	MIA	NYC	SFO	SEA	WAS
$w_{BOS,*}$	0	16578	4242	3365	22254	23665	205088	17165	4284	51895
$w_{DTT,*}$	16578	0	4448	5076	22463	24609	79945	13091	4172	19500
$w_{HOU,*}$	4242	4448	0	4370	17267	8602	28080	8455	2868	5616
$w_{MKC,*}$	3365	5076	4370	0	15287	4092	17291	8381	3033	7266
$w_{LAX,*}$	22254	22463	17267	15287	0	15011	105507	92083	32908	24583
$w_{MIA,*}$	23665	24609	8602	4092	15011	0	169397	8064	1840	20937
$w_{NYC,*}$	205088	79945	28080	17291	105507	169397	0	70935	14957	166694
$w_{SFO,*}$	17165	13091	8455	8381	92083	8064	70935	0	35285	19926
$w_{SEA,*}$	4284	4172	2868	3033	32908	1840	14957	35285	0	4951
$w_{WAS,*}$	51895	19500	5616	7266	24583	20937	166694	19926	4951	0
$c_{SFO,*}$	2703	2087	1650	1506	362	2591	2574	0.00	695	2430

Table 2: The normalized demands and normalized costs for the network in Table 1, given $\max_{i,j\in V} c_{ij} = 2725.79$.

	BOS	DTT	HOU	MKC	LAX	MIA	NYC	SFO	SEA	WAS	$w^{diff}_{SFO,*}$	$c^{norm}_{SFO,*}$
$\lvert w^{norm}_{BOS,*} - w^{norm}_{SFO,*}\rvert$	0.19	0.06	0.07	0.07	0.89	0.03	0.23	0.08	0.36	0.04	0.20	0.99
$\lvert w^{norm}_{DTT,*} - w^{norm}_{SFO,*}\rvert$	0.02	0.14	0.04	0.03	0.72	0.22	0.23	0.16	0.33	0.03	0.19	0.77
$\lvert w^{norm}_{HOU,*} - w^{norm}_{SFO,*}\rvert$	0.04	0.02	0.09	0.06	0.39	0.22	0.23	0.30	0.28	0.02	0.16	0.61
$\lvert w^{norm}_{MKC,*} - w^{norm}_{SFO,*}\rvert$	0.01	0.15	0.16	0.09	0.12	0.15	0.23	0.48	0.21	0.20	0.18	0.55
$\lvert w^{norm}_{LAX,*} - w^{norm}_{SFO,*}\rvert$	0.02	0.07	0.07	0.05	1.00	0.05	0.23	0.87	0.07	0.02	**0.25**	**0.13**
$\lvert w^{norm}_{MIA,*} - w^{norm}_{SFO,*}\rvert$	0.05	0.00	0.04	0.07	0.91	0.09	0.23	0.05	0.37	0.09	0.19	0.95
$\lvert w^{norm}_{NYC,*} - w^{norm}_{SFO,*}\rvert$	0.81	0.25	0.05	0.01	0.49	0.74	0.77	0.35	0.31	0.60	0.44	0.94
$\lvert w^{norm}_{SFO,*} - w^{norm}_{SFO,*}\rvert$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$\lvert w^{norm}_{SEA,*} - w^{norm}_{SFO,*}\rvert$	0.06	0.02	0.01	0.01	0.07	0.04	0.35	1.00	0.38	0.08	0.20	0.25
$\lvert w^{norm}_{WAS,*} - w^{norm}_{SFO,*}\rvert$	0.12	0.03	0.06	0.05	0.85	0.04	0.23	0.12	0.35	0.22	0.21	0.90

3.2. Contraction (Step 1)

According to Section 3.1, the contracted network is obtained by mapping nodes to contraction nodes. This contraction function should be designed based on both costs (c) and travel demands (w) between nodes. Intuitively, each node i tends to be mapped to the node j with smaller value of c_{ij}. In addition, if the distribution of $\{w_{ix}\}_{x\in V}$ and $\{w_{jx}\}_{x\in V}$ is similar with each other, node i and node j will also be more likely to be merged together, since the contracted network should keep the spatial properties and the distribution of travel demands in the original network. We normalize the travel demand (by computing the quotient of the largest demand from each node) and cost (by computing the quotient of the largest cost among the whole network) between each pair of nodes, as shown in Equations (46) and (47). The deviation of normalized demand between each pair of nodes is computed by Equation (48).

$$c^{norm}_{ij} = \frac{c_{ij}}{\max_{i,j\in V} c_{ij}} \tag{46}$$

$$w^{norm}_{ij} = \frac{w_{ij}}{\max_{x\in V} w_{ix}} \tag{47}$$

Algorithm 1 The contraction process by merging nodes

Input: The original network $G = (V, E)$ with the cost c_{ij} and the travel demand w_{ij} between each pair of nodes (i, j), cost coefficients $\alpha, \delta_1, \delta_2$, the size of the contracted network k.
Output: The contracted network $G^* = (V^*, E^*)$.
Let the current network be the original network, i.e., $G\_current = (V\_current, E\_current) = G = (V, E)$.
while $|V\_current| > k$ **do**
    Compute the value of w_{ij}^{diff} and c_{ij}^{norm} for each pair of nodes (i, j) for the current network according to Equations (46) and (48).
    Sort all node pairs by their values of $(w_{ij}^{diff} + c_{ij}^{norm})$ ascendingly in a list $Pairs$.
    Let S_{maxcap} be the set of top p nodes with the largest capacity in $V\_current$ (Only for capacitated HLPs).
    Let $Seen = \emptyset$ and $f\_current(i) = i, \forall i \in V\_current$.
    for $(i, j) \in Pairs$ **do**
        if $i \notin Seen$ and $j \notin Seen$ and $(i \notin S_{maxcap}$ or $j \notin S_{maxcap})$ **then**
            Use i to represent the node with larger value of $O_i + D_i$ in set $\{i, j\}$ and the node with the smaller value is represented by j (for capacitated HLPs, replace $O_i + D_i$ by $capacity_i$).
            Let $f\_current(j) = i$ and $Seen = Seen \cup \{i, j\}$.
            if $|Seen|/2 \geq |V\_current| - k$ **then**
                break
            end if
        end if
    end for
    Generate a new contracted network with the mapping function $f\_current$ on $G\_current = (V\_current, E\_current)$ according to Equation (44) and (45).
    Update the current network $G\_current = (V\_current, E\_current)$ with the newly generated network.
end while
Let the final contracted network be the current network, i.e., $G^* = (V^*, E^*) = G\_current = (V\_current, E\_current)$.

$$w_{ij}^{diff} = \frac{\sum_{x \in V} |w_{ix}^{norm} - w_{jx}^{norm}|}{n} \tag{48}$$

We use variables x_{ij} to represent the map function $f : V \to V$, i.e., $x_{ij} = 1$ if $f(i) = j$ and $x_{ij} = 0$ otherwise. The contraction process is formulated as follows:

$$\min \sum_{i \in V} \sum_{j \in V} \left(\theta_1 w_{ij}^{diff} + \theta_2 c_{ij}^{norm}\right) x_{ij} \tag{49}$$

$$\text{subject to} \sum_{j \in V} x_{ij} = 1, \forall i \in V \tag{50}$$

$$\sum_{j \in V} x_{jj} = k \tag{51}$$

$$x_{ij} \leq x_{jj}, \forall i, j \in V \tag{52}$$

$$x_{ij} \in \{0, 1\}, \forall i, j \in V \tag{53}$$

where θ_1 and θ_2 are the coefficients for normalizing two terms in the objective function. Based on the above formulation, we design a merging-based contraction strategy. By setting $\theta_1 = \theta_2 = 1$, node i tends to be merged with node j with smaller value of $(w_{ij}^{diff} + c_{ij}^{norm})$. For instance, the travel demands and costs between nodes in the CAB10 network are shown in Table 1. The travel demands between each pair of nodes and the costs from node SFO to all other nodes are given. Given $\max_{i,j \in V} c_{ij} = 2725.79$, the normalized demands and normalized costs are computed and the results are shown in Table 2. It is straightforward to compute:

$$w^{diff}_{SFO,LAX} + c^{norm}_{SFO,LAX} = 0.25 + 0.13 = 0.38$$

Therefore, node LAX is more likely to be merged with node SFO, compared to other nodes. By sorting all node pairs with their values of $(w^{diff}_{ij} + c^{norm}_{ij})$ in an ascending order and merging appropriate node pairs (in which both nodes are not in previously merged pairs), the contracted network is constructed. The pseudocode for this process is shown in Algorithm 1. Since nodes are merged pairwise, the size of the contracted network is at least half size of the original network in one contraction. In order to obtain a contracted network with a specific size k, one can perform contractions recursively (See Line 2 in Algorithm 1). In each iteration, after sorting all node pairs, an empty set $Seen$ and a self-mapping function $f\_current$ are generated (See Lines 3–6 in Algorithm 1). For each node pair in the list, if both nodes are not in the set $Seen$, then we map the node with smaller total demand $(O_i + D_i)$ to the other one (See Lines 7–10 in Algorithm 1). Note that for capacitated HLPs, each pair of nodes that both have the top p largest capacity should not be merged to avoid the infeasibility caused by contraction (See Lines 5,8–9 in Algorithm 1). Once k nodes are obtained, the merging operation is terminated (See Lines 11–13 in Algorithm 1). Based on the mapping function $f\_current$, the contracted network is generated. Then, in the next iteration, a new contraction will be performed on the currently generated network, until the network with the required size k is obtained (See Lines 17–18 in Algorithm 1). Given an original network with size n and a contraction size k, one needs $\lceil log_2(\frac{n}{k}) \rceil$ times of contraction to obtain the contracted network. Afterwards, the hub location problem on the contracted network is solved and the solutions are rewritten to the original network as the initial input.

3.3. Exploration (Step 2)

According to Section 3.1, the hub location problems on the contracted network can be solved by using any existing method. In this study, we use genetic algorithms and general variable neighborhood search to solve USApHMPC, CSApHMPC, and USApHMPI; and we use Benders decomposition to solve UMApHMPC, CMApHMPC and UMApHMPI. These methods have mostly been proposed in the literature for these HLPs or highly-similar ones; for the sake of self-containment and reproducibility, the major steps of these methods are summarized below.

3.3.1. Genetic algorithms

Genetic algorithms (GA) are a class of probabilistic algorithms in which solutions are encoded as chromosomes and evolutionary rules improve the fitness of a solution population. GA has been successfully used to solve several types of hub location problems, such as USApHMPC (Kratica et al. 2007), UMAHLPC (Kratica et al. 2005), and

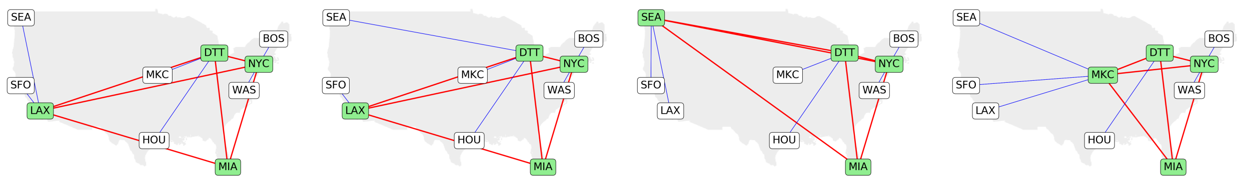


Figure 3: Several local search operations of general variable neighborhood search on USApHMPC. From left to right in the figure: The initial assignment, and operations Allocate, Alternate, and Locate. Hubs are highlighted in green color, hub links in red color, and spoke assignments in blue color.

CSApHMPC (Stanimirović 2012). Here, we introduce the process of GA with USApHMPC as an example; the process for other HLPs is rather similar. A solution is encoded by a pair of chromosomes, i.e., the hub set (*Hub*) and the node assignment ($Assignment$). We have $k \in Hub$ if $Y_{kk} = 1$ and $Assignment[i] = k$ if $Y_{ik} = 1$. The process of GA is summarized in the following steps Kratica et al. (2007):

1. **Initialization:** In this step, pn initial solutions are generated randomly by randomly selecting p hubs for each solution.
2. **Selection:** In each generation, several parents are selected to generate new offspring. In general, the solutions with larger fitness values are more likely to be selected. Therefore, a roulette wheel is used here.
3. **Crossover:** Given a pair of parents, both their hub sets and assignment lists are swapped. A reassignment for each spoke is performed to avoid the infeasibility of the offspring.
4. **Mutation:** For maintaining genetic diversity of the population, offspring undergo a mutation by randomly reassigning one or a pair of spoke nodes.

Elitism is applied after each iteration, discarding pn poor solutions. After a sufficient number of iterations, GA usually provides high-quality solutions for the problem. The above process is for USApHMPC. GA needs some modifications when solving other types of hub location problems.

1. For CSApHMPC, the structure of chromosome for each solution is the same with that for USApHMPC. However, the capacity constraint needs to be taken into account in the steps of initialization, crossover, and mutation.
2. For USApHMPI, in addition to *Hub* and $Assignment$, another list *Link* is required to denote the connected hub links, i.e., $(k, m) \in Links$ if $Z_{km} = 1$. To guarantee the feasibility of the solution, we also need to check the connection of hub networks in the steps of initialization and crossover. Mutation is fixed, since it only changes the assignment of spoke nodes.

3.3.2. *General variable neighborhood search*

General variable neighborhood search (GVNS) is a well-performed algorithm for solving single allocation hub location problems (Ilić et al. 2010). Starting from an initial solution, two phases are perfomed, i.e., the descent phase and the perturbation phase (Todosijević et al. 2017). In the descent phase, the algorithm finds the local minimum by searching and changing the neighborhood of the current solutions. In the perturbation phase, the algorithm gets out of the solution valley by a random shake operation. In this section, we introduce the process of GVNS with USApHMPC as an example. We first introduce several local search operations of GVNS (Ilić et al. 2010):

1. **Allocate:** In operation Allocate, the hub locations are not changed. For each spoke node, we try to assign it to another hub node. The operation with the largest reduction of cost is performed.
2. **Alternate:** This operation changes the hub locations. Before performing the operation, all nodes are grouped by p clusters, i.e., the sets of nodes that are assigned to the same hubs. For each cluster, we try to replace its hub by a spoke node in it. The operation with the largest reduction of cost is performed.
3. **Locate:** This operation changes the hub locations to increase the diversity of solutions. For each cluster, we try to replace its hub by a spoke node which is not in it. All nodes in this cluster are assigned to their closest hub nodes.
4. **Shake:** The above three operations are performed in the descent phase; while the Shake operation is performed in the perturbation phase to avoid local minimum solutions.

Based on the above operations, three types of general variable neighborhood search algorithms are designed for USApHMPC: sequential algorithm (Seq-GVNS), nested algorithm (Nest-GVNS), and mixed algorithm (Mix-GVNS) (Ilić et al. 2010). Seq-GVNS needs the shortest time but explores the smallest neighborhood. Nest-GVNS explores a large neighborhood, but its runtime is not acceptable. Mix-GVNS allows a trade-off between the size of neighborhood and runtime. Therefore, we use Mix-GVNS in this study. Similar with GA, GVNS also needs some modifications when solving other types of hub location problems.

1. For CSApHMPC, the capacity constraint needs to be considered when generating initial solutions and performing local search operations. The process for the former is the same with GA. In local search operations, when reassigning a spoke node to a new hub, only the hubs whose remaining capacities are larger than or equal to the out-flow of this spoke node can be the alternatives. When selecting a spoke node as a new hub, its capacity is also considered.
2. For USApHMPI, the hub links are also determined in the algorithm. There are three modifications in the process of GVNS: (1) In addition to hub locations and node assignments, the hub links are also determined when generating new solutions; (2) When replacing an old hub with a new hub node, the vertices of hub links are also replaced accordingly; (3) A new local search operation Reconnect is designed. In this operation, we remove an existing hub link and connect a pair of non-connected hub nodes. If the new hub network is still connected, i.e., there exist paths between each pair of hub nodes, this operation is called "feasible". Finally, the feasible operation with the largest reduction of cost is performed.

3.3.3. Benders decomposition

In Benders decomposition, the original problem is decomposed into a master problem and a sub-problem by keeping the values of some variables fixed (de Camargo et al. 2008, Contreras et al. 2011a). By solving the dual problem of the sub-problem, one or more new constraints (called Benders cuts) are added to the master problem in each iteration. The upper bound and the lower bound of the problem are updated accordingly. The algorithm terminates when the upper bound and the lower bound converge to the same value. The process of Benders decomposition for solving UMApHMPC is introduced as an example: Setting variables Y_k to fixed values $\hat{Y}_k$, we obtain the sub-problem (SP) and its dual problem (DSP). By solving the DSP, the Benders cut is constructed based on the obtained solution and added to the master problem (MP). The upper bound and lower bound are obtained by solving these problems in each iteration. The complete process of Benders decomposition for solving UMApHMPC is summarized below, which was proposed by de Camargo et al. (2008):

1. Set $UB = +\infty$ and $LB = 0$.
2. If $LB = UB$, terminate the algorithm. The optimal solution of the original problem is obtained.
3. Solve the MP and obtain the optimal values of the objective function $\hat{z}_{MP}$ and variables $\hat{Y}_k$.
4. Let $LB = \max(LB, \hat{z}_{MP})$. Update the values of $\hat{Y}_k$ in the new DSP.
5. Solve the new DSP. Obtain the optimal values of the objective function $\hat{z}_{DSP}$ and variables $\hat{\sigma}_{ij}, \hat{\pi}_{ijk}$.
6. Add the new Benders cut to the MP. Let $UB = \min(UB, \hat{z}_{DSP})$
7. Go back to Step 2.

3.4. Rewriting (Step 3)

By solving hub location problems in the contracted network, the obtained solutions induce the solutions to the original network by reassigning nodes in $V \backslash V^*$. Using USApHMPC as an example, let $Assignment$ represent the solution in the contracted network. If node i is allocated to hub k, then $Assignment[i] = k$. The rewriting step is formulated as follows:

$$\min \sum_{i \in V} \sum_{k \in V} c_{ik} Y_{ik} (\delta_1 O_i + \delta_2 D_i) + \sum_{i \in V} \sum_{k \in V} \sum_{m \in V} \alpha c_{km} X^i_{km} \tag{54}$$

$$\text{subject to } Y_{i,Assignment[i]} = 1, \forall i \in V^* \tag{55}$$

$$\sum_{k \in V} Y_{ik} = 1, \forall i \in V \tag{56}$$

$$\sum_{k \in V} Y_{kk} = p \tag{57}$$

$$Y_{ik} \leq Y_{kk}, \forall i, k \in V \tag{58}$$

$$\sum_{m\in V, m\neq k} X^i_{km} - \sum_{m\in V, m\neq k} X^i_{mk} = O_i Y_{ik} - \sum_{j\in V} w_{ij} Y_{jk}, \forall i,k \in V \tag{59}$$

$$Y_{ik} \in \{0,1\}, \forall i,k \in V \tag{60}$$

$$X^i_{km} \geq 0, \forall i,k,m \in V \tag{61}$$

The above formulation is a variant of USApHMPC with additional constraint (55). The new constraint makes sure that nodes in the contracted network must be assigned to their hubs in the contracted solution. The above formulation can be solved by different strategies. Since the rewritten solutions will be further improved in the next step *Optimization*, we use the simplest way to reassign the remaining nodes, i.e., assigning the nodes in $V \backslash V^*$ to their closest hubs. This rewriting strategy can be applied to three single allocation problems (USApHMPC, CSApHMPC, and USApHMPI). The rewritten solution for UMApHMPC, CMApHMPC, and UMApHMPI can be obtained by the hub sets (and sets of hub links) in the contracted network directly. Since the hub network is fully connected, we only need to select the best hub pair (k,m) with the minimum cost for transporting the travel demands between each pair of nodes (i, j).

3.5. *Optimization (Step 4)*

According to Section 3.1, the original hub location problems are solved by using the rewritten solutions as the initial input. We introduce several input strategies for different solution algorithms:

1. **Genetic Algorithms:** A type of algorithms based on population of solutions. Multiple initial solutions are required and multiple solutions are generated in each iteration. We need to select appropriate solutions (high-quality and high diversity) from a large solution space when solving the contracted problems and rewrite them back to the original network. The details for selecting these solutions are shown in Algorithm 2 and Appendix 6.2.

2. **General Variable Neighborhood Search:** This method requires only one initial solution and generates one solution in each iteration. When solving the contracted problem, we need to select the best solution with the minimum value of the objective function on the original network as the initial input solution. However, the best solution on the contracted network might not be the best solution on the original network. Therefore, we record all unique solutions in the contracted network and rewrite them. The best rewritten solution with the minimum value of the objective function on the original network is selected as the initial input for *Optimization* step. The details for this selection step are shown in Algorithm 3 and Appendix 6.2.

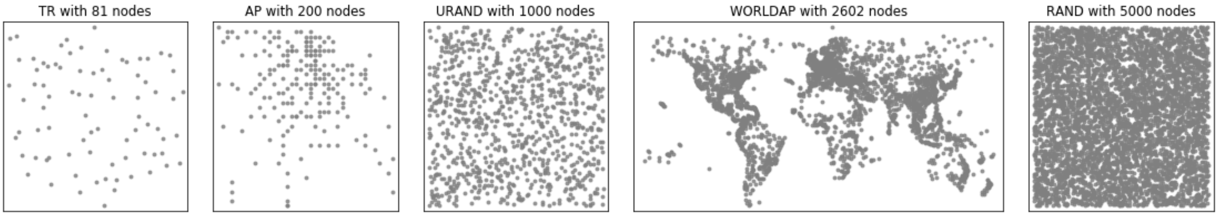


Figure 4: Visualizations of five evaluation datasets: TR dataset with 81 nodes, AP dataset with 200 nodes, URAND1000 dataset with 1000 nodes, WORLDAP dataset with 2602 nodes, and RAND5000 dataset with 5000 nodes.

3. **Benders decomposition:** Benders decomposition handles only one solution in each iteration. It selects the initial solution using the same strategy with GVNS. When resolving the original problem, the rewritten solution is used as the solution for the master problem (MP), i.e., Y_k. The value of Y_k is used to construct the dual sub-problem (DSP) for the first iteration.

4. Evaluation

4.1. Datasets and experimental setup

Three well-known datasets and two new datasets are used as case studies in our evaluation of EHLC. The TR (Turkish Postal) dataset includes 81 cities with pairwise distance and travel demands in the Turkish postal system (Çetiner 2003). The AP (Australia Post) dataset provides 200 postcode districts with locations and pairwise travel demands in Australia (Ernst and Krishnamoorthy 1996). The URAND1000 dataset is a random network with 1,000 nodes generated by Ilić et al. (2010). Two additional large datasets are generated for further evaluation of scalability. The WORLDAP dataset consists of 2,602 worldwide airports and actual travel demands between them, with ticket data including direct and indirect flights between airports. The data comes from Sabre Airport Data Intelligence (Sabre Airlines Solutions 2017). The RAND5000 dataset is generated by placing 5,000 nodes in a 1×1 uniformly plane at random, with travel demands randomly distributed in the range $[0, 1]$. All five datasets are visualized in Figure 4. The self-flows for nodes in the standard AP dataset were set to zero in our study. Because of the high complexity of CMApHMPC and UMApHMPI, we use TR40 dataset (which is generated by selecting the first 40

Table 3: Time cutoffs (in seconds) for different types of hub location problems, solution techniques, and datasets. A * indicates that the method was not tested on the dataset.

	TR40	TR	AP	URAND1000	WORLDAP	RAND5000
USApHMPC (GA and GVNS)	*	100	600	7,200	10,800	10,800
USApHMPI (GA and GVNS)	*	100	600	7,200	10,800	10,800
CSApHMPC (GA and GVNS)	*	100	600	7,200	10,800	10,800
UMApHMPC (Benders)	*	7,200	10,800	*	*	*
UMApHMPI (Benders)	1,200	*	*	*	*	*
CMApHMPC (Benders)	1,200	*	*	*	*	*

nodes from TR dataset, inspired by de Camargo et al. (2017)) for both problems. The cost coefficients of hub location problems are set to $\alpha \in \{0.3, 0.5, 0.7\}, \delta_1 = \delta_2 = 1$ for the TR/TR40 dataset, and $\alpha = 0.75, \delta_1 = 3, \delta_2 = 2$ for other datasets. For our major experiments, the contraction size is set as follows: we chose $k = 20$ for TR40, $k = 30$ for TR, $k = 50$ for AP, $k = 200$ for URAND1000, $k = 500$ for WORLDAP, and $k = 1000$ for RAND5000. These choices have been made with the goal of finding a trade-off between execution time (small contraction size) and quality (larger contraction size).

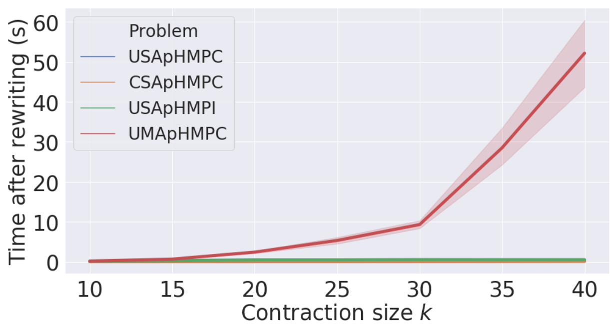
Given that we perform evaluations on six different HLPs, we have to restrict our implementation of state-of-the-art. For single allocation problems (USApHMPC, USApHMPI, and CSApHMPC), heuristics have been shown highly efficient in the literature. Therefore, GA (Kratica et al. 2007, Stanimirović 2012) and GVNS (Ilić et al. 2010)) are used to solve three types of single allocation problems on all datasets. Each instance of GA and GVNS is run with ten different seeds, because these algorithms are non-deterministic. For multiple allocation problems (UMApHMPC, CMApHMPC, and UMApHMPI), the performance of GA and GVNS is not as efficient as in single allocation problems. Therefore, Benders decomposition (de Camargo et al. 2009) is used to solve them. Given limited scalability of Benders decomposition, only the TR dataset and the AP dataset are used as case studies for UMApHMPC. Because of the high complexity of CMApHMPC and UMApHMPI, only the TR40 dataset (which is generated by selecting the first 40 nodes from TR dataset, inspired by de Camargo et al. (2017)) is used as a case study. Time cutoffs for different types of hub location problems, solution techniques, and datasets are shown in Table 3. These cutoff values were determined by initial sensitivity analysis considering the size of the datasets and the convergence of standard algorithms. All experiments were executed on a server with 40 cores and 430 GB RAM, running Fedora 26.

4.2. *Effect of the value of k*

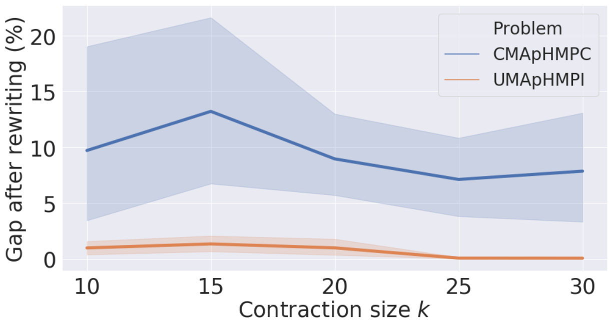
The value of k affects the final solution and the convergence speed of the algorithm significantly. Therefore, an evaluation on different values of k is performed in this subsection. Two datasets TR and TR40 (which is generated by selecting the first 40 nodes from TR dataset, inspired by de Camargo et al. (2017)) are selected as case studies. USApHMPC, CSApHMPC, USApHMPI, and UMApHMPC are solved on the TR dataset. CMApHMPC and UMApHMPI are solved on the TR40 dataset. The gaps of solutions and runtime until the rewriting step with different values of contraction size k are shown in Figure 5. Note that the gaps are obtained based on the best-known solutions.

For the TR dataset, $k = 30$ is an appropriate value which keep a trade-off between the qualities of the rewritten solutions and the runtime after the rewriting step. Although $k = 35$ further improves the solution qualities, the required runtime increases significantly as well. For the TR40 dataset, $k = 20$ is selected because of the same reason. For other datasets, we choose $k = 50$ for AP, $k = 200$ for URAND1000, $k = 500$ for WORLDAP, and $k = 1000$ for RAND5000.

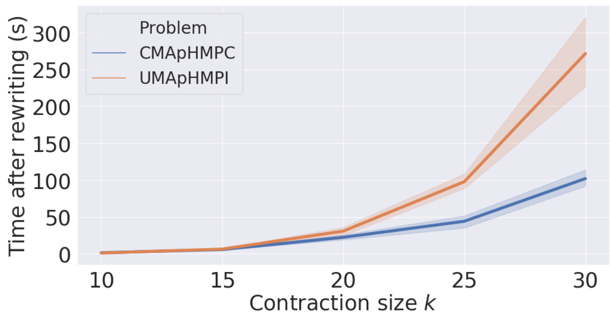
(a) Gaps on TR dataset.



(b) Runtime on TR dataset.



(c) Gaps on TR40 dataset.



(d) Runtime on TR40 dataset.

Figure 5: The gaps of solutions and runtime until the rewriting step with different values of contraction size k. USApHMPC, CSApHMPC, USApHMPI, and UMApHMPC are solved on the TR dataset. CMApHMPC and UMApHMPI are solved on the TR40 dataset.

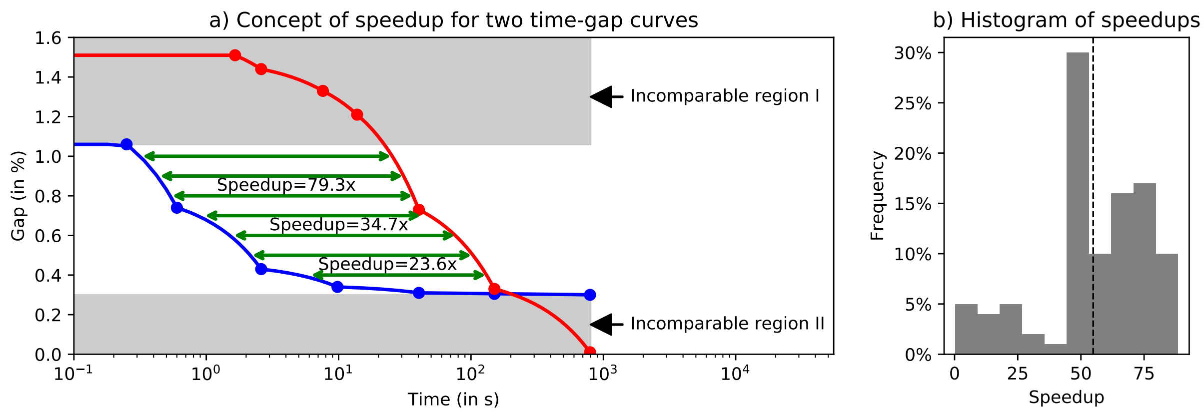


Figure 6: An example for two time-gap series and computing the speedups. Subfigure a) shows two time-gap series (one in blue color and one in red color). The speedups at different levels of gaps are indicated with green arrows. Two incomparable regions exist, highlighted in grey. Subfigure b) reports the frequency distribution of 100 equally-spaced sample speedups in the gap range 0.3 to 1.06. The median speedup is 54.85, as highlighted by the dashed line.

4.3. Execution time of EHLC compared to non-contraction methods

For our evaluation of EHLC, the major criterion of interest is the execution time for obtaining the same gaps as standard algorithms. Note that the gaps are obtained based on the best-known solutions. We design a comparison

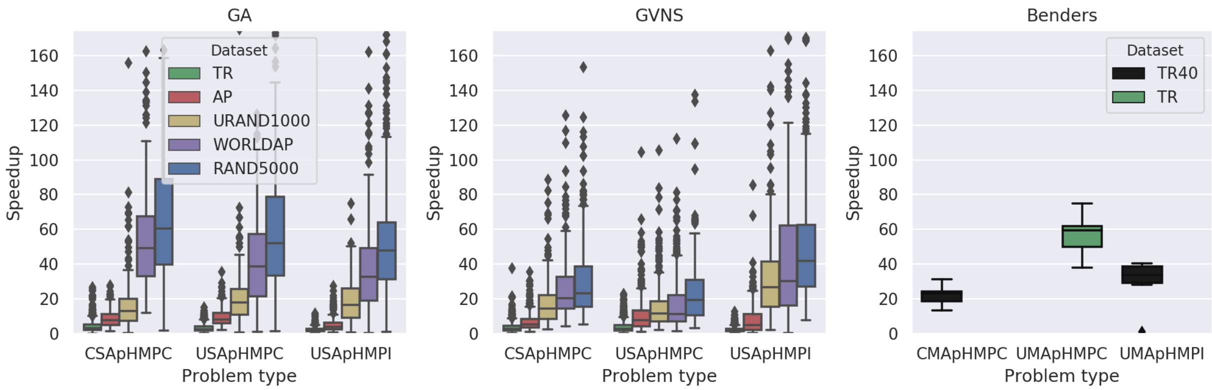


Figure 7: The median speedups for obtaining the same gaps by non-contraction methods and EHLC. GA and GVNS are applied for CSApHMPC, USApHMPC, and USApHMPI on the TR, AP, URAND1000, WORLDAP, and RAND5000 datasets. Benders decomposition is applied for UMApHMPC on the TR dataset and CMApHMPC, UMApHMPI on the TR40 dataset.

measure for two time-gap curves as visualized in Figure 6. The input consists of two time-gap curves as input, e.g., one from EHLC (blue) and one from non-contraction methods (red) for the same problem instance. The curve between two measure points, i.e., pairs of (time, gap), is interpolated using piecewise linear interpolation (Note that the x-axis in Figure 6 is log-scale). We identify the comparable region of both curves next, i.e., gap regions, for which both methods have obtained gap values. In the example, the region is between 0.3% and 1.04%. After discarding the incomparable regions, a collection of speedups can be obtained by uniform sampling along the gap axis: for each comparable gap value, we get two interpolated time values; dividing one (red) by the other (blue) yields a speedup. Computing the median speedup over all gap values gives an estimation of the overall speedup of EHLC (blue) versus non-contraction methods (red). Figure 6b) visualizes a histogram of all speedups for the given example, yielding a median speedup of 54.85.

We apply the described speedup estimation methodology for all competitors. The results are reported in Figure 7. We find that EHLC needs substantially shorter time than non-contraction methods for obtaining the same gaps in the vast majority of cases. For the largest datasets and more complicated problem formulations, such as USApHMPI and RAND5000, the median speedup of EHLC is 40x–50x, for both GA and GVNS. It should be noted that the exact solution technique, Benders decomposition, can be speed-up by a factor of 60x on UMApHMPC as well. Overall, we conclude that EHLC offers significant median speedups over all datasets and problem domains. Moreover, with increasing size of the network, the speedup grows substantially larger; a prerequisite for developing more scalable solution techniques for hub location problems. For further exploration of different speedup values, we analyze their relationship to the gap difference of EHLC after rewriting and initial solutions of non-contraction methods, i.e., we measure how much initial advantage we achieve by using the pivots from contraction. The results are reported in

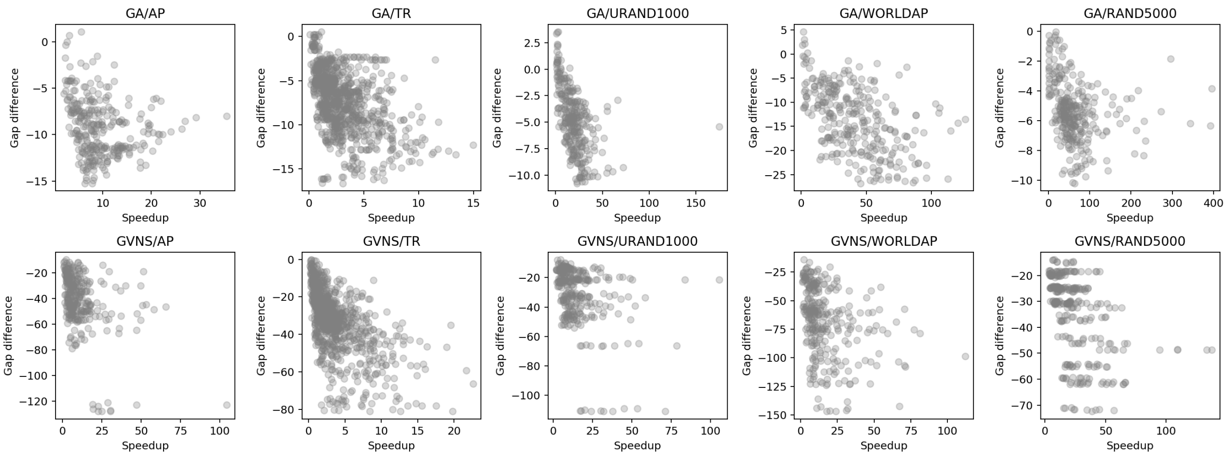


Figure 8: Correlation between gap difference and the speedup compared to non-contraction methods for USApHMPC. The gap difference is measured as the initial gap of EHLC (after rewriting step) minus the initial gap of non-contraction methods (based on their initial solution). Smaller initial gaps of EHLC lead to larger speedups.

Figure 8 for USApHMPC. It can be seen that, in most cases, the gap of EHLC after rewriting is much smaller compared to initial gaps of non-contraction methods; and larger deviations in the gap difference lead to substantial speedups. This speedup is much larger than the runtime used for solving the contracted problems. Appendix 6.4 contains additional charts for CSApHMPC and USApHMPI (See Figure 12 and Figure 13), confirming our results for USApHMPC.

4.4. The comparison on final gaps between EHLC and non-contraction methods

In the previous section, we have shown that EHLC can reach the same gaps as non-contraction methods significantly faster. In this subsection, we compare the final gaps obtained by EHLC and non-contraction methods (in the following abbreviated with *NC*). Table 4 provides an overview of min/median/max gaps for selected instances using GA and GVNS. The final min/median/max gaps of EHLC_GA and NC_GA are in the same magnitude, meaning that EHLC_GA usually finds solutions with similar costs as NC_GA. More important is the comparison of EHLC_GVNS with NC_GVNS, given that GVNS-based solutions clearly outperform GA-based solutions for all three HLPs. Again, EHLC_GVNS has similar min/median/max gaps as NC_GVNS, which means that EHLC does not sacrifice any amount of quality, given its tremendous speedup. More results can be found in Appendix 6.3; with same trends as those described above.

Figure 9 visualizes the final gaps of non-contraction solutions against the gaps of EHLC-based solutions as a scatterplot. Each data point stands for a combination of NC/EHLC for the same problem instances with different random seeds. For the majority of datasets, the overall gaps of both methods have a comparable order of magnitude. GVNS is better than GA for CSApHMPC and USApHMPC, and USApHMPI. Finally, Figure 10 visualizes the

Table 4: The minimum gaps, median gaps and maximum gaps of solutions (in %) obtained by EHLC_GA/GVNS and NC_GA/GVNS for USApHMPC/CSApHMPC/USApHMPI. The gaps that are smaller than or equal to 0.05% are highlighted in bold. NC=Non-contraction.

USApHMPC			**EHLC_GA**			**NC_GA**			**EHLC_GVNS**			**NC_GVNS**		
Dataset	α	p	Min	Med	Max	Min	Med	Max	Min	Med	Max	Min	Med	Max
TR	0.5	4	0.65	0.65	2.09	**0.00**	0.11	0.98	**0.00**	0.10	0.77	**0.00**	**0.03**	0.11
TR	0.5	8	1.30	1.47	1.89	0.33	0.93	1.94	**0.00**	0.36	0.37	**0.00**	0.20	0.52
AP	0.75	4	0.63	0.63	0.84	**0.00**	0.01	0.48	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
AP	0.75	8	0.27	0.82	2.22	1.00	2.24	4.26	**0.00**	0.06	0.26	**0.00**	**0.00**	0.26
URAND1000	0.75	4	0.20	0.56	1.81	0.30	0.81	1.44	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
URAND1000	0.75	8	2.03	3.72	4.57	2.07	3.71	5.73	**0.00**	**0.01**	**0.02**	**0.00**	**0.00**	0.12
WORLDAP	0.75	4	0.88	2.37	5.77	0.48	3.07	4.73	**0.00**	**0.01**	0.06	**0.00**	**0.02**	3.77
WORLDAP	0.75	8	3.01	5.22	11.41	3.34	6.63	14.14	**0.03**	2.28	5.25	**0.00**	3.71	9.08
RAND5000	0.75	4	0.44	1.06	2.02	0.64	0.85	2.09	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
RAND5000	0.75	8	1.77	3.74	4.82	2.84	3.78	5.85	**0.00**	**0.03**	0.20	**0.00**	0.16	0.63
CSApHMPC			**EHLC_GA**			**NC_GA**			**EHLC_GVNS**			**NC_GVNS**		
Dataset	α	p	Min	Med	Max	Min	Med	Max	Min	Med	Max	Min	Med	Max
TR	0.5	4	0.07	0.07	1.89	**0.00**	0.07	0.35	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
TR	0.5	8	0.89	0.89	1.41	**0.03**	0.64	2.07	**0.03**	**0.03**	0.46	**0.00**	**0.03**	0.46
AP	0.75	4	0.31	0.31	1.92	**0.00**	0.38	1.72	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
AP	0.75	8	0.35	0.35	1.32	0.52	1.64	3.82	**0.00**	0.46	1.25	**0.00**	0.12	1.00
URAND1000	0.75	4	1.10	1.65	3.32	0.57	2.37	7.76	**0.00**	**0.02**	**0.03**	**0.00**	**0.02**	0.18
URAND1000	0.75	8	2.37	6.16	8.85	2.65	5.73	8.65	0.09	0.21	0.66	**0.00**	0.22	0.58
WORLDAP	0.75	4	0.75	3.12	4.74	1.26	4.80	8.64	**0.00**	**0.00**	**0.02**	**0.00**	**0.03**	2.84
WORLDAP	0.75	8	2.91	6.89	13.59	8.95	15.19	29.91	**0.00**	3.35	7.70	1.19	2.87	4.78
RAND5000	0.75	4	1.89	2.75	4.63	1.66	2.74	5.02	**0.00**	**0.05**	0.38	**0.00**	0.10	0.37
RAND5000	0.75	8	6.06	7.61	12.99	8.32	10.68	15.76	0.22	1.17	4.52	**0.00**	1.22	2.73
USApHMPI			**EHLC_GA**			**NC_GA**			**EHLC_GVNS**			**NC_GVNS**		
Dataset	α	p,q	Min	Med	Max	Min	Med	Max	Min	Med	Max	Min	Med	Max
TR	0.5	4,5	0.62	0.64	2.30	0.09	0.64	1.18	**0.00**	**0.04**	0.10	**0.04**	0.10	0.74
TR	0.5	8,12	2.00	3.20	4.77	0.85	3.22	5.05	**0.00**	0.52	2.08	0.07	0.21	0.66
AP	0.75	4,5	0.64	0.83	0.96	0.11	0.46	5.47	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
AP	0.75	8,12	0.36	2.16	4.94	2.01	5.66	9.18	**0.00**	0.17	1.84	**0.00**	0.24	1.84
URAND1000	0.75	4,5	0.48	1.33	2.10	0.80	1.40	2.83	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.01**
URAND1000	0.75	8,12	3.64	7.51	10.77	5.53	7.84	10.20	**0.03**	0.13	1.28	**0.00**	0.28	2.63
WORLDAP	0.75	4,5	0.69	1.48	6.86	0.80	3.81	6.06	**0.00**	**0.01**	0.30	**0.00**	0.06	2.94
WORLDAP	0.75	8,12	8.77	12.25	16.24	6.62	13.13	19.58	**0.00**	3.32	6.24	0.90	7.08	15.11
RAND5000	0.75	4,5	0.60	1.55	2.26	1.25	1.68	3.09	**0.00**	**0.03**	**0.04**	**0.02**	**0.03**	0.18
RAND5000	0.75	8,12	4.93	8.09	9.13	7.51	8.91	12.24	**0.00**	0.11	1.56	**0.02**	0.84	1.63

dependency between initial gaps and final gaps of methods. It can be seen that the initial gaps of EHLC (obtained after rewriting) are significantly smaller than those of standard methods. Moreover, initial gaps of NC_GA are smaller than those of NC_GVNS, given the larger random solution population as input.

The gaps of solutions for UMApHMPC, CMApHMPC, and UMApHMPI obtained by EHLC_Benders decomposition and NC_Benders decomposition are shown in Tables 8–10 in Appendix 6.3. Both methods provide the (nearly) optimal solutions for the TR dataset and TR40 dataset. However, only EHLC_Benders decomposition solves the prob-

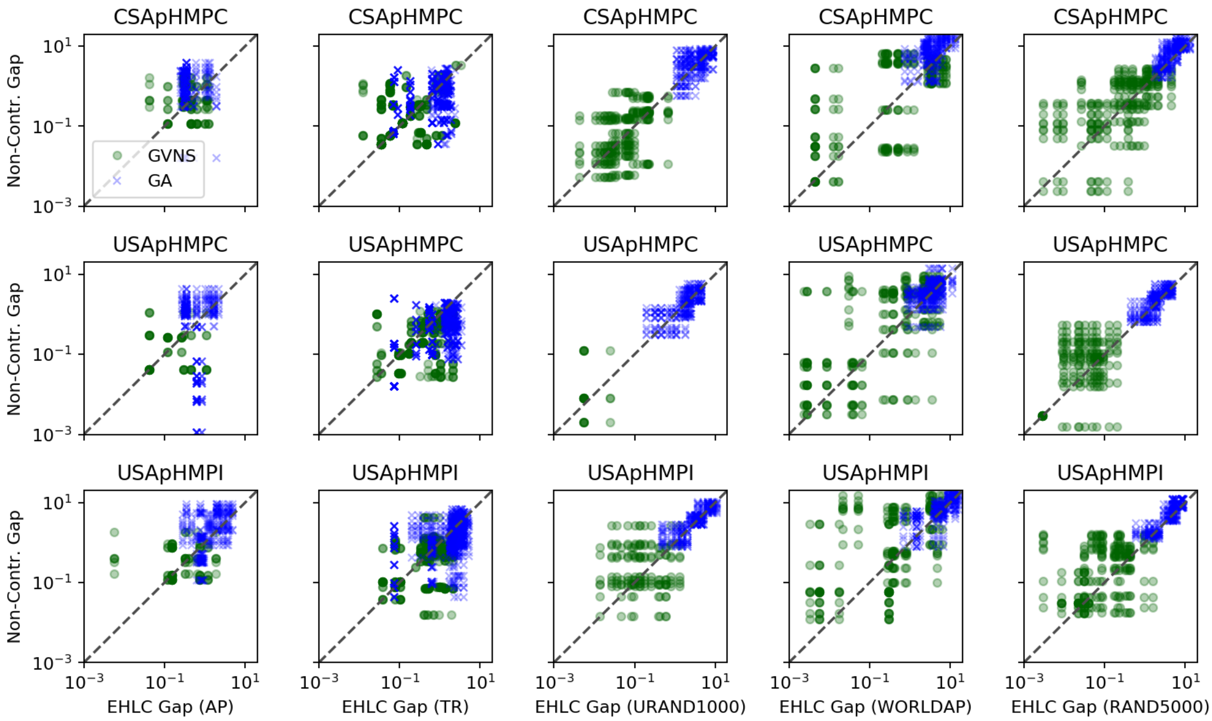


Figure 9: Scatterplot of gaps obtained by non-contraction methods vs gaps obtained by EHLC; results for GA and GVNS are distinguished by color (GA=blue and GVNS=green). Note that both axes are log scaled and best-known solutions (with gap 0%) are not shown.

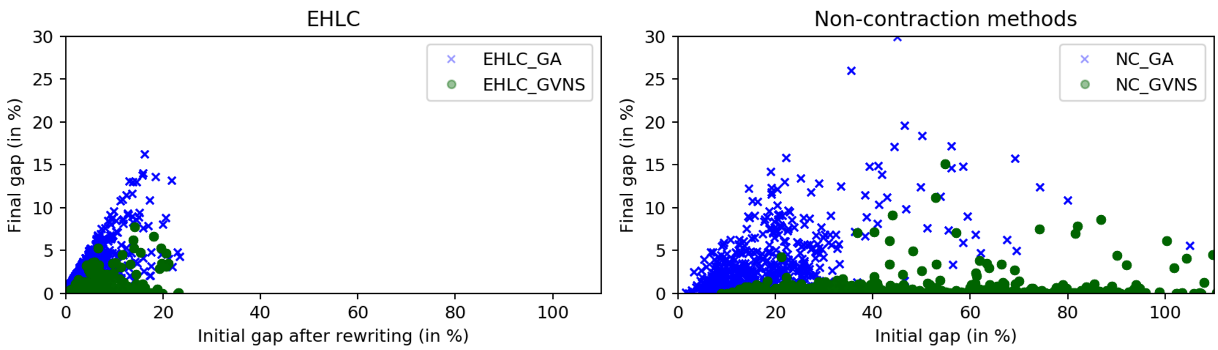
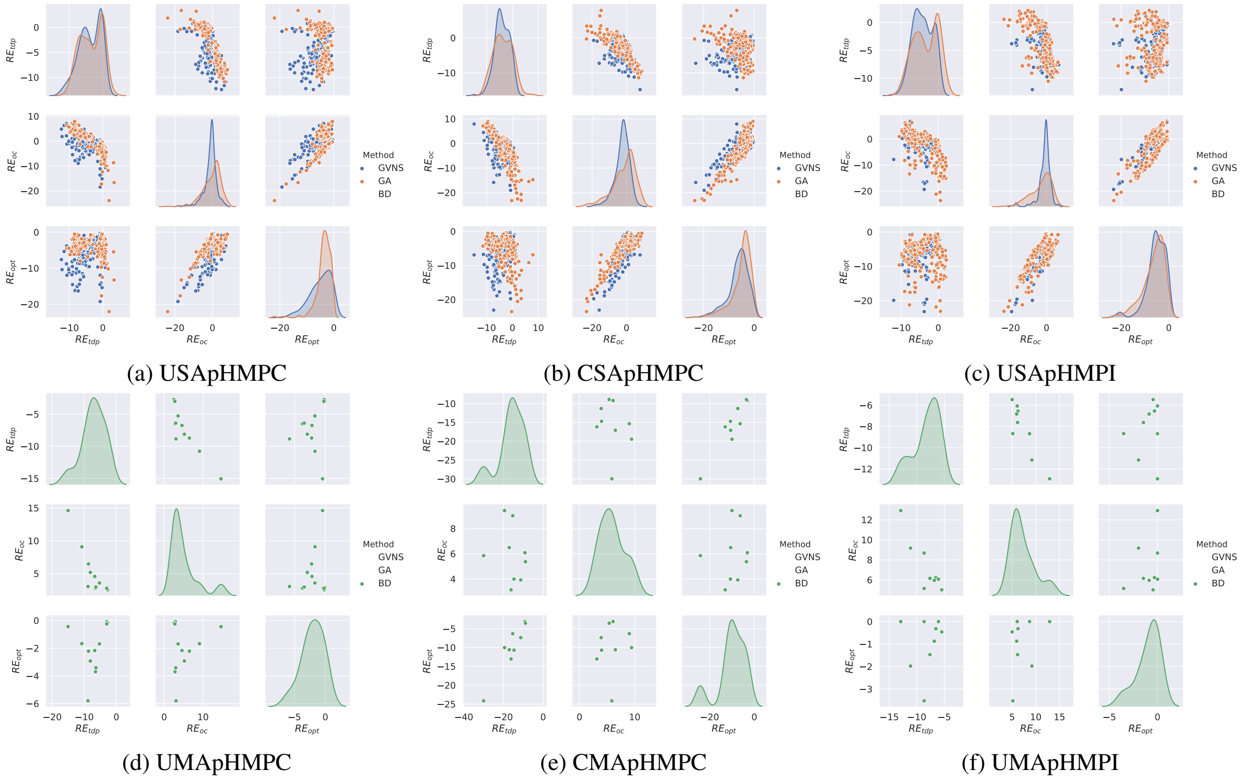


Figure 10: Scatterplot of initial gaps vs. final gaps obtained by EHLC and non-contraction methods. The initial gaps of EHLC (obtained after rewriting) are significantly smaller than those of standard methods. GA has substantially better initial solutions than GVNS, given the initial population of 200 individuals.

lems on the AP dataset towards optimality. The reason is that the high-quality initial solutions obtained by contraction speeds up the convergence of Benders decomposition significantly, so that it can solve the problem optimally within the given limited time on the AP dataset.



(a) USApHMPC (b) CSApHMPC (c) USApHMPI

(d) UMApHMPC (e) CMApHMPC (f) UMApHMPI

Figure 11: The pairwise distribution of the total error, opportunity cost error and optimality error of the contraction strategy on different types of hub location problems.

4.5. Error analysis on the contraction strategy

As discussed in Section 2, several error measurements for demand point aggregation (i.e., node contraction) have been proposed. In this subsection, we select three of them and calculate their (approximate) values based on the experimental results in the above subsections. The selected error measurements are total demand point error, (approximate) opportunity cost error and (approximate) optimality error. We first explain the definitions of these errors.

Assume that the original network and the contracted network are represented by G and G^*, respectively. Let $F(X, G)$ and $F(X, G^*)$ represent the models based on corresponding networks, where X is the solution which is mainly determined by the locations of hub nodes. We assume that the optimal solutions of $F(X, G)$ and $F(X, G^*)$ are $\bar{X}$ and X^*, respectively. The selected error measurements are defined as follows:

$$\text{Total demand point error:} \qquad E_{tdp} = F(X^*, G^*) - F(X^*, G) \tag{62}$$

$$\text{Opportunity cost error:} \qquad E_{oc} = F(\bar{X}, G) - F(X^*, G^*) \tag{63}$$

$$\text{Optimality error:} \qquad E_{opt} = F(\bar{X}, G) - F(X^*, G) \tag{64}$$

It would be very interesting to provide a bound which is tight enough for the error, i.e., $|f(X, G^*) - f(X, G)| < EB, \forall X$. Francis and Lowe (1993) proposed that the error bound of the multi-facility minisum model is $\sum_{i,j} w_{ij} c_{i,f(i)}$, where node i is contracted to node $f(i)$. Therefore, the error bound of USApHMPC is $\sum_{i \in V}(\delta_1 O_i c_{i,f(i)} + \delta_2 D_i c_{f(i),i})$. Other HLPs have corresponding formulations. Apart from few special datasets in which nodes are distributed in strong clusters, this error bound could be very loose. This is another reason why we only compute the practical values of the above three error measures.

The total demand point error can be computed easily and indicates the quality of the approximating function based on the contracted network G^* with solution X^* (Francis et al. 2009). When computing the latter two error measurements, the optimal solution $\bar{X}$ cannot be obtained easily (Otherwise, we donot need to do the contraction for solving the original problem). Therefore, we use the best obtained solution for each instance in Sections 4.3–4.4 as the approximate value of $\bar{X}$. The solution obtained in the contracted network (See step Exploration in Section 3.3) is used as the approximate value of X^*. To show standard values for different types of hub location problems on various datasets, we normalize the above error measurements by computing their ratios as follows:

$$\text{Ratio of total demand point error:} \qquad RE_{tdp} = \frac{F(X^*, G^*) - F(X^*, G)}{F(\bar{X}, G)} \tag{65}$$

$$\text{Ratio of opportunity cost error:} \qquad RE_{oc} = \frac{F(\bar{X}, G) - F(X^*, G^*)}{F(\bar{X}, G)} \tag{66}$$

$$\text{Ratio of optimality error:} \qquad RE_{opt} = \frac{F(\bar{X}, G) - F(X^*, G)}{F(\bar{X}, G)} \tag{67}$$

The pairwise distribution of RE_{tdp}, RE_{oc} and RE_{opt} for USApHMPC, CSApHMPC, USApHMPI, UMApHMPC, CMApHMPC, and UMApHMPI on different datasets are shown in Figure 11. The results obtained by GVNS, GA and Benders decomposition are represented by blue, orange and green dots. There are some observations as follows:

1. Because of the contraction/aggregation of the travel demands, the values of the objective functions are smaller in the contracted network compared to the original network. Therefore, the ratio of total demand point error $RE_{tdp} < 0$ in most cases. The other two error measurements are also usually negative because of the high quality of $\bar{X}$.

2. There is a significantly positive correlation between RE_{oc} and RE_{opt}, i.e., the quality of solution X^* on the contracted network is highly correlated to that on the original network. It indicates that the contraction strategy performs well on preserving the properties of the original network on hub location problems.

5. Conclusions

There is a need to develop scalable algorithms for hub location problems, given that the computation of exact solutions to these problems is usually NP-hard. We propose a novel view on speeding up standard solution techniques by the notion of contraction (termed EHLC), which exploits the structural similarities between original networks and contracted/condensed subnetworks. Using the solutions obtained by contracted networks as input to solvers for the original network, the solution algorithms can speed up significantly, with few problem-specific modifications.

We have conducted and reported the results of experiments regarding six standard hub location problems (USApHMPC, CSApHMPC, USApHMPI, UMApHMPC, CMApHMPC, and UMApHMPI) and several standard solution techniques (GA, GVNS, and Benders decomposition). We show that EHLC achieves similar gaps as the standard solution techniques, but significantly faster. For the largest datasets in our study, WORLDAP and RAND5000, median speedups of 40x–50x were achieved, compared to GA and GVNS. These results not only hold for constructing feasible solutions, but also for solving HLPs towards optimality, as our experiments with Benders decomposition for UMApHMPC, CMApHMPC, and UMApHMPI show, with a median speedup of 20x–60x. The speedup of EHLC grows with the size of the input network, allowing to compute solutions 1–2 orders of magnitude faster. Our contribution pushes the state-of-the-art boundary on hub location solutions techniques.

There are several directions for future work, based on the results of this study. First, we believe that the concept underlying EHLC can be adapted to many more types of HLPs. Second, experiments on even larger networks could lead towards further speedups; making it feasible to solve HLPs at unprecedented scale. Third, EHLC is designed as one-directional, where the contracted network is used at the initial phase of the solution process. It could be beneficial to blend the process of solving in contracted and original network further.

References

Abyazi-Sani, R., Ghanbari, R., 2016. An efficient tabu search for solving the uncapacitated single allocation hub location problem. Computers & Industrial Engineering 93, 99 – 109. URL: http://www.sciencedirect.com/science/article/pii/S0360835215005082, doi:https://doi.org/10.1016/j.cie.2015.12.028.

Alumur, S., Kara, B., Karasan, O., 2009. The design of single allocation incomplete hub networks. Transportation Research Part B: Methodological 43, 936–951.

An, Y., Zhang, Y., Zeng, B., 2015. The reliable hub-and-spoke design problem: Models and algorithms. Transportation Research Part B: Methodological 77, 103–122.

Azizi, N., Chauhan, S., Salhi, S., Vidyarthi, N., 2016. The impact of hub failure in hub-and-spoke networks: Mathematical formulations and solution techniques. Computers & Operations Research 65, 174–188.

Brimberg, J., Mladenović, N., Todosijević, R., Urošević, D., 2017. General variable neighborhood search for the uncapacitated single allocation p-hub center problem. Optimization Letters 11, 377–388. URL: https://doi.org/10.1007/s11590-016-1004-x, doi:10.1007/s11590-016-1004-x.

Calık, H., Alumur, S., Kara, B.Y., Karasan, O.E., 2009. A tabu-search based heuristic for the hub covering problem over incomplete hub networks. Computers & Operations Research 36, 3088 – 3096. URL: http://www.sciencedirect.com/science/article/pii/S0305054808002487, doi:https://doi.org/10.1016/j.cor.2008.11.023. new developments on hub location.

de Camargo, R.S., de Miranda, G., Luna, H.P.L., 2008. Benders decomposition for the uncapacitated multiple allocation hub location problem. Computers & Operations Research 35, 1047–1064.

de Camargo, R.S., de Miranda, G., Luna, H.P.L., 2009. Benders decomposition for hub location problems with economies of scale. Transportation Science 43, 86–97. URL: http://dx.doi.org/10.1287/trsc.1080.0233, doi:10.1287/trsc.1080.0233.

de Camargo, R.S., de Miranda, G., O'Kelly, M.E., Campbell, J.F., 2017. Formulations and decomposition methods for the incomplete hub location network design problem with and without hop-constraints. Applied Mathematical Modelling 51, 274 – 301. URL: http://www.sciencedirect.com/science/article/pii/S0307904X17304274, doi:https://doi.org/10.1016/j.apm.2017.06.035.

Campbell, J., 1994. Integer programming formulations of discrete hub location problems. European Journal of Operational Research 72, 387–405.

Campbell, J., 1996. Hub location and the p-hub median problem. Operations Research 44, 923–935.

Campbell, J., 2009. Hub location for time definite transportation. Computers & Operations Research 36, 3107–3116.

Campbell, J., Ernst, A.T., Krishnamoorthy, M., 2005. Hub arc location problems: Part i—introduction and results. Management Science 51, 1540–1555. doi:10.1287/mnsc.1050.0406.

Campbell, J., O'Kelly, M.E., 2012. Twenty-five years of hub location research. Transportation Science 46, 153–169.

Campbell, J.F., 1993. Continuous and discrete demand hub location problems. Transportation Research Part B: Methodological 27, 473 – 482. URL: http://www.sciencedirect.com/science/article/pii/0191261593900186, doi:https://doi.org/10.1016/0191-2615(93)90018-6.

Campbell, J.F., Stiehr, G., Ernst, A.T., Krishnamoorthy, M., 2003. Solving hub arc location problems on a cluster of workstations. Parallel Computing 29, 555 – 574. URL: http://www.sciencedirect.com/science/article/pii/S0167819103000425, doi:https://doi.org/10.1016/S0167-8191(03)00042-5. parallel computing in logistics.

Çetiner, S., 2003. An iterative hub location and routing problem for postal delivery systems. Ph.D. thesis. Citeseer.

Contreras, I., Cordeau, J.F., Laporte, G., 2011a. Benders decomposition for large-scale uncapacitated hub location. Operations research 59, 1477–1490.

Contreras, I., Díaz, J.A., Fernández, E., 2009a. Lagrangean relaxation for the capacitated hub location problem with single assignment. OR Spectrum 31, 483–505. URL: https://doi.org/10.1007/s00291-008-0159-y, doi:10.1007/s00291-008-0159-y.

Contreras, I., Díaz, J.A., Fernández, E., 2011b. Branch and price for large-scale capacitated hub location problems with single assignment. INFORMS Journal on Computing 23, 41–55.

Contreras, I., Fernández, E., Marín, A., 2009b. Tight bounds from a path based formulation for the tree of hub location problem. Computers & Operations Research 36, 3117–3127.

Dai, W., Zhang, J., Sun, X., Wandelt, S., 2017. General contraction method for uncapacitated single allocation p-hub median problems, in: 2017 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–8. doi:10.1109/SSCI.2017.8285321.

Dai, W., Zhang, J., Sun, X., Wandelt, S., 2019. HUBBI: Iterative network design for incomplete hub location problems. Computers & Operations Research 104, 394 – 414. URL: http://www.sciencedirect.com/science/article/pii/S0305054818302521, doi:https://doi.org/10.1016/j.cor.2018.09.011.

Ebery, J., Krishnamoorthy, M., Ernst, A., Boland, N., 2000. The capacitated multiple allocation hub location problem: Formulations and algorithms. European Journal of Operational Research 120, 614–631.

Emir-Farinas, H., Francis, R.L., 2005. Demand point aggregation for planar covering location models. Annals of Operations Research 136, 175–192. URL: https://doi.org/10.1007/s10479-005-2044-2, doi:10.1007/s10479-005-2044-2.

Ernst, A.T., Krishnamoorthy, M., 1996. Efficient algorithms for the uncapacitated single allocation p-hub median problem. Location science 4, 139–154.

Ernst, A.T., Krishnamoorthy, M., 1998. Exact and heuristic algorithms for the uncapacitated multiple allocation p-hub median problem. European Journal of Operational Research 104, 100 – 112. URL: http://www.sciencedirect.com/science/article/pii/S0377221796003402, doi:https://doi.org/10.1016/S0377-2217(96)00340-2.

Ernst, A.T., Krishnamoorthy, M., 1999. Solution algorithms for the capacitated single allocation hub location problem. Annals of Operations Research 86, 141–159. URL: https://doi.org/10.1023/A:1018994432663, doi:10.1023/A:1018994432663.

Farahani, R.Z., Hekmatfar, M., Arabani, A.B., Nikbakhsh, E., 2013. Hub location problems: A review of models, classification, solution techniques, and applications. Computers & Industrial Engineering 64, 1096–1109.

Figueiredo, R.M.A., O'Kelly, M.E., Pizzolato, N.D., 2014. A two-stage hub location method for air transportation in Brazil. International Transactions in Operational Research 21, 275–289.

Francis, R.L., Lowe, T., 1993. On worst-case aggregation analysis for network location problems. Annals of Operations Research 40, 229–246. doi:10.1007/BF02060479.

Francis, R.L., Lowe, T.J., Rayco, M.B., Tamir, A., 2009. Aggregation error for location models: survey and analysis. Annals of Operations Research 167, 171–208. URL: https://doi.org/10.1007/s10479-008-0344-z, doi:10.1007/s10479-008-0344-z.

Francis, R.L., Lowe, T.J., Tamir, A., 2002. Demand point aggregation for location models, in: Facility location. Springer, Berlin, pp. 207–232.

Francis, R.L., Lowe, T.J., Tamir, A., Emir-Farinas, H., 2004. A framework for demand point and solution space aggregation analysis for location models. European Journal of Operational Research 159, 574 – 585. URL: http://www.sciencedirect.com/science/article/pii/S0377221703004338, doi:https://doi.org/10.1016/S0377-2217(03)00433-8.

Gelareh, S., Nickel, S., 2011. Hub location problems in transportation networks. Transportation Research Part E: Logistics and Transportation Review 47, 1092–1111.

Ghaffarinasab, N., Kara, B.Y., 2019. Benders decomposition algorithms for two variants of the single allocation hub location problem. Networks and Spatial Economics 19, 83–108. URL: https://doi.org/10.1007/s11067-018-9424-z, doi:10.1007/s11067-018-9424-z.

Hoff, A., Peiró, J., Corberán, Á., Martí, R., 2017. Heuristics for the capacitated modular hub location problem. Computers & Operations Research 86, 94 – 109. URL: http://www.sciencedirect.com/science/article/pii/S0305054817301144, doi:https://doi.org/10.1016/j.cor.2017.05.004.

Ilić, A., Urošević, D., Brimberg, J., Mladenović, N., 2010. A general variable neighborhood search for solving the uncapacitated single allocation p-hub median problem. European Journal of Operational Research 206, 289–300.

Karimi, H., 2018. The capacitated hub covering location-routing problem for simultaneous pickup and delivery systems. Computers & Industrial Engineering 116, 47 – 58. URL: http://www.sciencedirect.com/science/article/pii/S0360835217305879, doi:https://doi.org/10.1016/j.cie.2017.12.020.

Kim, H., O'Kelly, M.E., 2009. Reliable p-hub location problems in telecommunication networks. Geographical Analysis 41, 283–306.

Kratica, J., Stanimirović, Z., Tošić, D., Filipović, V., 2007. Two genetic algorithms for solving the uncapacitated single allocation p-hub median problem. European Journal of Operational Research 182, 15–28.

Kratica, J., Stanimirović, Z., Tošić, D., Filipović, V., 2005. Genetic algorithm for solving uncapacitated multiple allocation hub location problem. Computing and Informatics 24, 427–440.

Labbé, M., Yaman, H., Gourdin, E., 2005. A branch and cut algorithm for hub location problems with single assignment. Mathematical Programming 102, 371–405. URL: https://doi.org/10.1007/s10107-004-0531-x, doi:10.1007/s10107-004-0531-x.

Meier, J.F., Clausen, U., 2018. Solving single allocation hub location problems on euclidean data. Transportation Science 52, 1141–1155.

O'Kelly, M.E., 1986. The location of interacting hub facilities. Transportation science 20, 92–106.

O'Kelly, M.E., 1987. A quadratic integer program for the location of interacting hub facilities. European Journal of Operational Research 32, 393–404.

O'Kelly, M.E., 1992. A clustering approach to the planar hub location problem. Annals of Operations Research 40, 339–353.

O'Kelly, M.E., 2012. Fuel burn and environmental implications of airline hub networks. Transportation Research Part D: Transport and Environment 17, 555–567.

Peker, M., Kara, B., Campbell, J., Alumur, S., 2015. Spatial analysis of single allocation hub location problems. Networks and Spatial Economics , 1–27.

Pirkul, H., Schilling, D., 1998. An efficient procedure for designing single allocation hub and spoke systems. Management Science 44, S235–S242.

Plastria, F., 2001. On the choice of aggregation points for continuous p-median problems: A case for the gravity centre. TOP: An Official Journal of the Spanish Society of Statistics and Operations Research 9, 217–242. doi:10.1007/BF02579084.

Rodríguez-Martín, I., Salazar-González, J.J., Yaman, H., 2014. A branch-and-cut algorithm for the hub location and routing problem. Computers & Operations Research 50, 161–174.

Rodríguez-Martín, I., Salazar-González, J.J., 2008. Solving a capacitated hub location problem. European Journal of Operational Research 184, 468 – 479. URL: http://www.sciencedirect.com/science/article/pii/S0377221706011647, doi:https://doi.org/10.1016/j.ejor.2006.11.026.

Rogers, D., Plante, R., Wong, R., Evans, J., 1991. Aggregation and disaggregation techniques and methodology in optimization. Operations Research 39. doi:10.1287/opre.39.4.553.

Sabre Airlines Solutions, 2017. Airport Data Intelligence. URL: http://www.sabreairlinesolutions.com .

Sadeghi, M., Jolai, F., Tavakkoli-Moghaddam, R., Rahimi, Y., 2015. A new stochastic approach for a reliable p-hub covering location problem. Computers & Industrial Engineering 90, 371 – 380. URL: http://www.sciencedirect.com/science/article/pii/S0360835215003964, doi:https://doi.org/10.1016/j.cie.2015.10.003.

Silva, M.R., Cunha, C.B., 2009. New simple and efficient heuristics for the uncapacitated single allocation hub location problem. Computers & Operations Research 36, 3152 – 3165. URL: http://www.sciencedirect.com/science/article/pii/S0305054809000033, doi:https://doi.org/10.1016/j.cor.2008.12.019. new developments on hub location.

Skorin-Kapov, D., Skorin-Kapov, J., 1994. On tabu search for the location of interacting hub facilities. European Journal of Operational Research 73, 502–509.

Stanimirović, Z., 2012. A genetic algorithm approach for the capacitated single allocation p-hub median problem. Computing and Informatics 29, 117–132.

de Sá, E.M., Morabito, R., de Camargo, R.S., 2018. Benders decomposition applied to a robust multiple allocation incomplete hub location problem. Computers & Operations Research 89, 31 – 50. URL: http://www.sciencedirect.com/science/article/pii/S0305054817302009, doi:https://doi.org/10.1016/j.cor.2017.08.001.

Taherkhani, G., Alumur, S.A., 2019. Profit maximizing hub location problems. Omega 86, 1 – 15. URL: http://www.sciencedirect.com/science/article/pii/S030504831731068X, doi:https://doi.org/10.1016/j.omega.2018.05.016.

Thomadsen, T., Larsen, J., 2007. A hub location problem with fully interconnected backbone and access networks. Computers & Operations Research 34, 2520 – 2531. URL: http://www.sciencedirect.com/science/article/pii/S0305054805003096, doi:https://doi.org/10.1016/j.cor.2005.09.018.

Todosijević, R., Urošević, D., Mladenović, N., Hanafi, S., 2017. A general variable neighborhood search for solving the uncapacitated r-allocation p-hub median problem. Optimization Letters 11, 1109–1121. URL: https://doi.org/10.1007/s11590-015-0867-6, doi:10.1007/s11590-015-0867-6.

Yaman, H., Carello, G., 2005. Solving the hub location problem with modular link capacities. Computers & Operations Research 32, 3227–3245.

Yang, K., Liu, Y., Yang, G., 2013. An improved hybrid particle swarm optimization algorithm for fuzzy p-hub center problem. Computers & Industrial Engineering 64, 133 – 142. URL: http://www.sciencedirect.com/science/article/pii/S0360835212002227, doi:https://doi.org/10.1016/j.cie.2012.09.006.

6. Appendix

6.1. The sub-problems and master problems of Benders decomposition for UMApHPMC, CMApHMPC and UMApHMPI

For the reproduction of Benders decomposition algorithm on UMApHPMC, CMApHMPC and UMApHMPI, we introduce the sub-problems and master problems of Benders decomposition for these problems.

UMApHMPC: The Benders decomposition for solving UMApHMPC and similar hub location problems were proposed by de Camargo et al. (2008, 2009).

Set variables Y_k to fixed values $\hat{Y}_k$ in Equations (21–26), the sub-problem (SP) is generated:

$$\min \sum_{i\in V}\sum_{j\in V}\sum_{k\in V}\sum_{m\in V}\left(\delta_1 c_{ik} + \alpha c_{km} + \delta_2 c_{mj}\right) w_{ij} X_{ijkm} \tag{68}$$

$$\text{subject to} \sum_{k\in V}\sum_{m\in V} X_{ijkm} = 1, \forall i, j \in V \tag{69}$$

$$\sum_{m\in V} X_{ijkm} + \sum_{m\in V, m\neq k} X_{ijmk} \leq \hat{Y}_k, \forall i, j, k \in V \tag{70}$$

$$X_{ijkm} \in \{0, 1\}, \forall i, j, k, m \in V \tag{71}$$

Let $\sigma_{ij} \in R$ and $\pi_{ijk} \geq 0$ be the dual variables corresponding to Equations (69–70), the dual sub-problem (DSP) is generated:

$$\max \sum_{i\in V}\sum_{j\in V}(\sigma_{ij} - \sum_{k\in V}\hat{Y}_k \pi_{ijk}) \tag{72}$$

$$\text{subject to } \sigma_{ij} - \pi_{ijk} - \pi_{ijm} \leq (\delta_1 c_{ik} + \alpha c_{km} + \delta_2 c_{mj}) w_{ij}, \forall i, j, k, m \neq k \in V \tag{73}$$

$$\sigma_{ij} - \pi_{ijk} \leq (\delta_1 c_{ik} + \delta_2 c_{kj}) w_{ij}, \forall i, j, k \in V \tag{74}$$

$$\sigma_{ij} \in R, \forall i, j \in V \tag{75}$$

$$\pi_{ijk} \geq 0, \forall i, j, k \in V \tag{76}$$

Using θ_{ij} to represent the cost for OD pair (i, j), the master problem (MP) is formulated as follows:

$$\min \sum_{i\in V}\sum_{j\in V}\theta_{ij} \tag{77}$$

$$\text{subject to} \sum_{k\in V} Y_k = p \tag{78}$$

$$Y_k \in \{0, 1\}, \forall k \in V \tag{79}$$

In each iteration, after obtaining the values of $\hat{\sigma}_{ij}$ and $\hat{\pi}_{ijk}$ by solving the DSP, the Benders cuts are generated and added to MP for the next iteration.

$$\theta_{ij} \geq \hat{\sigma}_{ij} - \sum_{k \in V} \hat{\pi}_{ijk} Y_k, \forall i, j \in V \tag{80}$$

CMApHMPC: We did not find the appropriate references about Benders decomposition for CMApHMPC. Therefore, we design this algorithm by ourselves, inspired by de Camargo et al. (2008).

Set variables Y_k to fixed values $\hat{Y}_k$ in Equations (21–27), the sub-problem (SP) is generated:

$$\min \sum_{i \in V} \sum_{j \in V} \sum_{k \in V} \sum_{m \in V} \left(\delta_1 c_{ik} + \alpha c_{km} + \delta_2 c_{mj}\right) w_{ij} X_{ijkm} \tag{81}$$

$$\text{subject to} \sum_{k \in V} \sum_{m \in V} X_{ijkm} = 1, \forall i, j \in V \tag{82}$$

$$\sum_{m \in V} X_{ijkm} + \sum_{m \in V, m \neq k} X_{ijmk} \leq \hat{Y}_k, \forall i, j, k \in V \tag{83}$$

$$\sum_{i \in V} \sum_{j \in V} \sum_{m \in V} X_{ijkm} w_{ij} \leq \lambda_k \hat{Y}_k, \forall k \in V \tag{84}$$

$$X_{ijkm} \in \{0, 1\}, \forall i, j, k, m \in V \tag{85}$$

Let $\sigma_{ij} \in R$, $\pi_{ijk} \geq 0$ and $\eta_k \geq 0$ be the dual variables corresponding to Equations (82–84), the dual sub-problem (DSP) is generated:

$$\max \sum_{i \in V} \sum_{j \in V} (\sigma_{ij} - \sum_{k \in V} \hat{Y}_k \pi_{ijk}) - \sum_{k \in V} \lambda_k \hat{Y}_k \eta_k \tag{86}$$

$$\text{subject to } \sigma_{ij} - \pi_{ijk} - \pi_{ijm} - w_{ij} \eta_k \leq (\delta_1 c_{ik} + \alpha c_{km} + \delta_2 c_{mj}) w_{ij}, \forall i, j, k, m \neq k \in V \tag{87}$$

$$\sigma_{ij} - \pi_{ijk} - w_{ij} \eta_k \leq (\delta_1 c_{ik} + \delta_2 c_{kj}) w_{ij}, \forall i, j, k \in V \tag{88}$$

$$\sigma_{ij} \in R, \forall i, j \in V \tag{89}$$

$$\pi_{ijk} \geq 0, \forall i, j, k \in V \tag{90}$$

$$\eta_k \geq 0, \forall k \in V \tag{91}$$

Using θ to represent the cost for all OD pairs, the master problem (MP) is formulated as follows:

$$\min \ \theta \tag{92}$$

$$\text{subject to } \sum_{k\in V} Y_k = p \tag{93}$$

$$Y_k \in \{0,1\}, \forall k \in V \tag{94}$$

In each iteration, after obtaining the values of $\hat{\sigma}_{ij}$ and $\hat{\pi}_{ijk}$ by solving the DSP, the Benders cuts are generated and added to MP for the next iteration.

$$\theta \geq \sum_{i\in V}\sum_{j\in V}(\hat{\sigma}_{ij} - \sum_{k\in V}\hat{\pi}_{ijk}Y_k) - \sum_{k\in V}\lambda_k Y_k \hat{\eta}_k, \text{ if DSP is optimally solved} \tag{95}$$

$$0 \geq \sum_{i\in V}\sum_{j\in V}(\hat{\sigma}_{ij} - \sum_{k\in V}\hat{\pi}_{ijk}Y_k) - \sum_{k\in V}\lambda_k Y_k \hat{\eta}_k, \text{ if DSP is unbounded} \tag{96}$$

UMApHMPI: The Benders decomposition for solving UMApHMPI was proposed by de Camargo et al. (2017). Set variables Y_k, Z_{km} to fixed values $\hat{Y}_k, \hat{Z}_{km}$ in Equations (28–43), the sub-problem (SP) is generated:

$$\min \sum_{i\in V}\sum_{j\in V} w_{ij}\left[\sum_{k\in V}\delta_1 c_{ik}h_{ijk} + \sum_{m\in V}\delta_2 c_{mj}t_{ijm} + \sum_{k\in V}\sum_{m\in V}\alpha c_{km}X_{ijkm}\right] \tag{97}$$

$$\sum_{m\in V, m\neq j} t_{ijm} + h_{ijj} + \sum_{k\in V, k\neq j} X_{ijkj} = 1, \forall i,j \in V \tag{98}$$

$$h_{ijm} + \sum_{k\in V, k\neq j, k\neq m} X_{ijkm} = \sum_{k\in V, k\neq i, k\neq m} X_{ijmk} + t_{ijm}, \forall i,j,m \in V, i\neq m, j\neq m \tag{99}$$

$$t_{iji} + \sum_{m\in V, m\neq i} X_{ijim} = \hat{Y}_i, \forall i,j \in V \tag{100}$$

$$h_{ijk} + \sum_{m\in V, m\neq j, m\neq k} X_{ijmk} \leq \hat{Y}_k, \forall i,j,k \in V, k\neq i, k\neq j \tag{101}$$

$$h_{ijj} + \sum_{k\in V, k\neq j} X_{ijkj} = \hat{Y}_j, \forall i,j \in V \tag{102}$$

$$X_{ijkm} \leq (\hat{Z}_{km} \text{ if } k > m) + (\hat{Z}_{mk} \text{ if } k < m), \forall i,j,k,m \in V, k\neq j, m\neq i, m\neq k \tag{103}$$

$$X_{ijkm} \in \{0,1\}, \forall i,j,k,m \in V \tag{104}$$

$$h_{ijk} \in \{0,1\}, \forall i,j,k \in V \tag{105}$$

$$t_{ijk} \in \{0,1\}, \forall i,j,k \in V \tag{106}$$

Let $\theta_{ij} \in R, \gamma_{ijk} \in R, \delta_{ij} \in R, \phi_{ijk} \geq 0, \beta_{ij} \in R, \tau_{ijkm} \geq 0$ be the dual variables corresponding to Equations (98–103), the dual sub-problem (DSP) is generated:

$$\max \sum_{i\in V}\sum_{j\in V}\left[\theta_{ij} - \hat{Y}_i\delta_{ij} - \hat{Y}_j\beta_{ij} - \sum_{k\in V,k\neq i,k\neq j}\hat{Y}_k\phi_{ijk} - \sum_{k\in V,k\neq j}\left(\sum_{m\in V,m\neq i,m<k}\hat{Z}_{km}\tau_{ijkm} + \sum_{m\in V,m\neq i,m>k}\hat{Z}_{mk}\tau_{ijkm}\right)\right] \quad (107)$$

$$\text{subject to } \theta_{ij} - \beta_{ij} \leq \delta_1 c_{ij}w_{ij}, \forall i,j \in V \quad (108)$$

$$\gamma_{ijk} - \phi_{ijk} \leq \delta_1 c_{ik}w_{ij}, \forall i,j,k \in V, k\neq i, k\neq j \quad (109)$$

$$\gamma_{ijk} - \delta_{ijk} - \phi_{ijk} - \tau_{ijik} \leq \alpha c_{ik}w_{ij}, \forall i,j,k \in V, k\neq i, k\neq j \quad (110)$$

$$\gamma_{ijm} - \gamma_{ijk} - \phi_{ijm} - \tau_{ijkm} \leq \alpha c_{km}w_{ij}, \forall i,j,k,m \in V, k\neq i, k\neq j, m\neq i, m\neq j, k\neq m \quad (111)$$

$$\theta_{ij} - \gamma_{ijm} - \beta_{ij} - \tau_{ijmj} \leq \alpha c_{mj}w_{ij}, \forall i,j,m \in V, m\neq i, m\neq j \quad (112)$$

$$\theta_{ij} - \delta_{ij} - \beta_{ij} - \tau_{ijij} \leq \alpha c_{ij}w_{ij}, \forall i,j \in V \quad (113)$$

$$\theta_{ij} - \delta_{ij} \leq \delta_2 c_{ij}w_{ij}, \forall i,j \in V \quad (114)$$

$$\theta_{ij} - \gamma_{ijm} \leq \delta_2 cmjw_{ij}, \forall i,j,m \in V, m\neq i, m\neq j \quad (115)$$

$$\theta_{ij} \in R, \forall i,j \in V \quad (116)$$

$$\gamma_{ijk} \in R, \forall i,j,k \in V, k\neq i, k\neq j \quad (117)$$

$$\delta_{ij} \in R, \forall i,j \in V \quad (118)$$

$$\phi_{ijk} \geq 0, \forall i,j,k \in V, k\neq i, k\neq j \quad (119)$$

$$\beta_{ij} \in R, \forall i,j \in V \quad (120)$$

$$\tau_{ijkm} \geq 0, \forall i,j,k,m \in V, k\neq j, m\neq i, k\neq m \quad (121)$$

Using ψ_{ij} to represent the cost for each OD pair (i, j), the master problem (MP) is formulated as follows:

$$\min \sum_{i\in V}\sum_{j\in V}\psi_{ij} \quad (122)$$

$$\text{subject to } \sum_{k\in V} Y_k = p \quad (123)$$

$$\sum_{k\in V} Y_k = p \quad (124)$$

$$\sum_{k\in V}\sum_{m\in V,m<k} Z_{km} = q \quad (125)$$

$$Z_{km} \leq Y_k, \forall k, m<k \in V \quad (126)$$

$$Z_{km} \leq Y_m, \forall k, m<k \in V \quad (127)$$

$$Z_{km} \in \{0,1\}, \forall k, m \in V, m < k \tag{128}$$

$$Y_k \in \{0,1\}, \forall k \in V \tag{129}$$

In each iteration, after obtaining the values of dual variables by solving the DSP, the Benders cuts are generated and added to MP for the next iteration.

$$\psi_{ij} \geq \hat{\theta}_{ij} - Y_i\hat{\delta}_{ij} - Y_j\hat{\beta}_{ij} - \sum_{k\in V, k\neq i, k\neq j} Y_k\hat{\phi}_{ijk} - \sum_{k\in V, k\neq j}\left(\sum_{m\in V, m\neq i, m<k} Z_{km}\hat{\tau}_{ijkm} + \sum_{m\in V, m\neq i, m>k} Z_{mk}\hat{\tau}_{ijkm}\right), \text{ if DSP is optimally solved} \tag{130}$$

$$0 \geq \hat{\theta}_{ij} - Y_i\hat{\delta}_{ij} - Y_j\hat{\beta}_{ij} - \sum_{k\in V, k\neq i, k\neq j} Y_k\hat{\phi}_{ijk} - \sum_{k\in V, k\neq j}\left(\sum_{m\in V, m\neq i, m<k} Z_{km}\hat{\tau}_{ijkm} + \sum_{m\in V, m\neq i, m>k} Z_{mk}\hat{\tau}_{ijkm}\right), \text{ if DSP is unbounded} \tag{131}$$

6.2. The selection of rewritten solutions for different solution algorithms

According to Section 3.1, the solutions on the contracted network are rewritten back to the original network. Several of these rewritten solutions are selected and the original hub location problem is optimized with them as the initial input. However, some solution algorithms need multiple initial solutions and generate multiple new solutions in each iteration, while other solution algorithms deal with a single solution every time. Therefore, different selection strategies should be designed for different types of solution algorithms:

1. **For population-based algorithms, such as genetic algorithms:** Multiple initial solutions are required and multiple solutions are generated in each iteration. Therefore, multiple solutions are recorded and rewritten from the contracted network. The key point is to keep the diversity of solutions. The pseudocode in Algorithm 2 is used to explain the selection strategy for this type of algorithms.

   As shown in Line 1, an empty set $All\_Solution$ is generated initially. Each unique solution in the initial population or generated in each iteration is recorded in this set (See Lines 2–14). All solutions in the set $All\_Solution$ are sorted by the values of their objective functions ascendingly (See Line 15). A new empty set $Selected\_Solution$ is generated afterwards (See Line 16). Each solution in $All\_Solution$ with the hub set appearing less than a given number of times in $Selected\_Solution$ will be added to $Selected\_Solution$. The loop is terminated until $Selected\_Solution$ reaches the size of $Init\_Population$.

2. **For single-solution algorithms, such as variable neighborhood search algorithm and Benders decomposition:** The algorithms start from one initial solution and generate only one solution in each iteration. The best solution obtained by the rewriting phase is selected in this case. The selection strategy for this type of algorithms is simpler

Algorithm 2 The strategy for selecting contracted solutions for population-based algorithms.

Input: The contracted network $G^* = (V^*, E^*)$, the original network $G = (V, E)$, the set of initial solutions $Init\_Population$ for the contracted problem and the maximum times T of repeating hub sets.
Output: The set of rewritten solutions.
Let $All\_Solution = \emptyset$ be the set to record all unique solutions.
for each $solution \in Init\_Population$ **do**
  if $solution \notin All\_Solution$ **then**
    Let $All\_Solution = All\_Solution \cup \{solution\}$.
  end if
end for
for each iteration of the solution algorithm **do**
  Let $New\_Solution$ be the set of newly generated solutions.
  for each $solution \in New\_Solution$ **do**
    if $solution \notin All\_Solution$ **then**
      Let $All\_Solution = All\_Solution \cup \{solution\}$.
    end if
  end for
end for
Sort the solutions in $All\_Solution$ by the values of their objective functions ascendingly.
Let $Selected\_Solution = \emptyset$ be the set to record all selected solutions.
for each $solution \in All\_Solution$ **do**
  if the hub set of $solution$ appears less than T times in $Selected\_Solution$ **then**
    Let $Selected\_Solution = Selected\_Solution \cup \{solution\}$.
  end if
  if $|Selected\_Solution| == |Init\_Population|$ **then**
    break
  end if
end for
Rewriting all solutions in $Selected\_Solution$ to the original network.

Algorithm 3 The strategy for selecting contracted solutions for single-solution algorithms.

Input: The contracted network $G^* = (V^*, E^*)$, the original network $G = (V, E)$, and the initial solution $init\_solu$ for the contracted problem.
Output: The selected rewritten solution.
1: Let $All\_Solution = \{init\_solu\}$ be the set to record all unique solutions.
2: **for** each iteration of the solution algorithm **do**
3:   Let new_solu be newly generated solution.
4:   **if** $new\_solu \notin All\_Solution$ **then**
5:     Let $All\_Solution = All\_Solution \cup \{new\_solu\}$.
6:   **end if**
7: **end for**
8: Rewriting all solutions in $All\_Solution$ to the original network.
9: Select the rewritten solution with the lowest objective function value on the original network.

than the former one. As shown in Algorithm 3, an initial set $All\_Solution$ which contains the initial solution is generated in Line 1. Each unique solution which is generated in each iteration is added to the set $All\_Solution$ (See Lines 2–7). After rewriting all solutions in $All\_Solution$, the solution with the lowest objective function value on the original network is selected.

6.3. *The detailed experimental results for USApHMPC, CSApHMPC, USApHMPI, UMApHMPC, CMApHMPC, and UMApHMPI.*

Table 5: The minimum gaps, median gaps and maximum gaps of solutions (in %) obtained by EHLC_GA/GVNS and NC_GA/GVNS as well as the median values and the maximum values of the median speedups (MSU) between EHLC and NC methods for USApHMPC. The numbers of instances that $Median\_Gap_{EHLC\_GA} \leq Median\_Gap_{NC\_GA}$ and $Median\_Gap_{EHLC\_GVNS} \leq Median\_Gap_{NC\_GVNS}$ are 9 and 13 out of 21, respectively. The gaps that are smaller than or equal to 0.05% are highlighted in bold.

USApHMPC			**EHLC_GA Gap**			**EHLC_GA MSU**		**NC_GA Gap**			**EHLC_GVNS Gap**			**EHLC_GVNS MSU**		**NC_GVNS Gap**		
Dataset	α	p	Min	Med	Max	Med	Max	Min	Med	Max	Min	Med	Max	Med	Max	Min	Med	Max
TR	0.3	4	0.07	0.07	0.07	3.75	11.52	**0.00**	0.08	2.45	**0.00**	**0.00**	**0.00**	6.48	19.78	**0.00**	**0.00**	**0.00**
TR	0.3	6	0.26	0.53	2.61	2.55	7.52	**0.00**	0.17	1.72	**0.00**	**0.00**	0.17	3.65	22.61	**0.00**	**0.00**	0.17
TR	0.3	8	0.55	0.55	1.35	4.14	12.71	0.13	0.60	1.52	**0.00**	0.32	0.61	2.63	16.26	**0.00**	**0.00**	0.67
TR	0.5	4	0.65	0.65	2.09	0.82	6.59	**0.00**	0.11	0.98	**0.00**	0.10	0.77	2.91	17.51	**0.00**	**0.03**	0.11
TR	0.5	6	1.60	2.12	2.80	2.31	7.61	0.07	0.54	1.88	**0.00**	0.71	1.08	2.14	5.78	**0.00**	0.52	0.71
TR	0.5	8	1.30	1.47	1.89	5.82	14.96	0.33	0.93	1.94	**0.00**	0.36	0.37	3.26	19.53	**0.00**	0.20	0.52
TR	0.7	4	1.83	1.83	2.30	1.36	4.79	**0.00**	0.67	0.69	**0.00**	**0.00**	**0.00**	2.49	5.85	**0.00**	**0.00**	**0.00**
TR	0.7	6	2.52	2.52	3.39	1.65	3.88	**0.00**	0.27	1.47	**0.03**	1.02	2.05	1.22	6.43	**0.00**	0.28	1.02
TR	0.7	8	1.56	1.76	2.50	3.51	7.09	0.15	0.90	1.74	**0.00**	0.47	1.63	2.45	16.06	**0.00**	**0.01**	1.22
AP	0.75	4	0.63	0.63	0.84	11.18	21.63	**0.00**	**0.01**	0.48	**0.00**	**0.00**	**0.00**	12.09	104.38	**0.00**	**0.00**	**0.00**
AP	0.75	6	0.32	0.34	1.69	5.45	21.04	0.49	1.19	1.75	**0.00**	**0.04**	1.11	7.85	65.69	**0.00**	**0.04**	1.13
AP	0.75	8	0.27	0.82	2.22	7.98	35.53	1.00	2.24	4.26	**0.00**	0.06	0.26	5.19	33.75	**0.00**	**0.00**	0.26
URAND1000	0.75	4	0.20	0.56	1.81	24.30	72.42	0.30	0.81	1.44	**0.00**	**0.00**	**0.00**	16.32	105.56	**0.00**	**0.00**	**0.00**
URAND1000	0.75	6	1.33	1.95	2.60	15.12	175.14	0.87	1.84	4.07	**0.00**	**0.00**	**0.00**	11.02	58.34	**0.00**	**0.00**	**0.05**
URAND1000	0.75	8	2.03	3.72	4.57	14.79	49.95	2.07	3.71	5.73	**0.00**	**0.01**	**0.02**	8.82	36.22	**0.00**	**0.00**	0.12
WORLDAP	0.75	4	0.88	2.37	5.77	35.84	126.22	0.48	3.07	4.73	**0.00**	**0.01**	**0.06**	12.09	81.36	**0.00**	**0.02**	3.77
WORLDAP	0.75	6	1.18	3.06	6.67	25.70	73.72	1.32	2.72	3.96	**0.00**	0.39	3.46	11.03	112.16	**0.00**	1.74	4.04
WORLDAP	0.75	8	3.01	5.22	11.41	52.91	112.76	3.34	6.63	14.14	**0.03**	2.28	5.25	10.27	64.97	**0.00**	3.71	9.08
RAND5000	0.75	4	0.44	1.06	2.02	49.61	189.67	0.64	0.85	2.09	**0.00**	**0.00**	0.00	29.46	137.66	**0.00**	**0.00**	**0.00**
RAND5000	0.75	6	1.40	1.87	2.54	48.26	232.21	1.51	2.45	3.15	**0.00**	**0.03**	0.09	13.72	44.31	**0.02**	0.07	0.20
RAND5000	0.75	8	1.77	3.74	4.82	57.38	395.42	2.84	3.78	5.85	**0.00**	**0.03**	0.20	14.91	56.78	**0.00**	0.16	0.63

Table 6: The minimum gaps, median gaps and maximum gaps of solutions (in %) obtained by EHLC_GA/GVNS and NC_GA/GVNS as well as the median values and the maximum values of the median speedups (MSU) between EHLC and NC methods for CSApHMPC. The numbers of instances that $Median\_Gap_{EHLC\_GA} \leq Median\_Gap_{NC\_GA}$ and $Median\_Gap_{EHLC\_GVNS} \leq Median\_Gap_{NC\_GVNS}$ are 12 and 17 out of 21, respectively. The gaps that are smaller than or equal to 0.05% are highlighted in bold.

CSApHMPC			**EHLC_GA Gap**			**EHLC_GA MSU**		**NC_GA Gap**			**EHLC_GVNS Gap**			**EHLC_GVNS MSU**		**NC_GVNS Gap**		
Dataset	α	p	Min	Med	Max	Med	Max	Min	Med	Max	Min	Med	Max	Med	Max	Min	Med	Max
TR	0.3	4	0.07	0.07	0.07	0.53	9.97	**0.00**	**0.04**	1.63	**0.00**	**0.00**	**0.00**	2.54	10.57	**0.00**	**0.00**	**0.00**
TR	0.3	6	0.09	0.09	2.35	4.28	20.73	**0.00**	**0.00**	2.50	**0.00**	**0.00**	3.31	2.88	13.69	**0.00**	**0.00**	3.31
TR	0.3	8	0.18	0.18	0.89	5.28	26.54	0.18	0.38	1.34	**0.00**	0.18	0.30	3.03	37.60	**0.00**	0.18	0.30
TR	0.5	4	0.07	0.07	1.89	1.89	6.83	**0.00**	0.07	0.35	**0.00**	**0.00**	**0.00**	2.65	9.23	**0.00**	**0.00**	**0.00**
TR	0.5	6	0.66	0.66	1.27	4.04	13.75	0.12	0.78	2.06	**0.00**	0.12	2.39	2.91	15.58	**0.00**	0.12	0.12
TR	0.5	8	0.89	0.89	1.41	5.12	16.92	**0.03**	0.64	2.07	**0.03**	**0.03**	0.46	4.00	20.70	**0.00**	**0.03**	0.46
TR	0.7	4	1.39	1.50	1.65	2.51	7.79	**0.00**	0.96	2.67	0.07	0.32	0.34	2.01	9.15	0.07	0.34	0.35
TR	0.7	6	1.06	1.40	1.91	2.61	10.83	0.14	0.32	1.48	**0.00**	0.52	1.41	2.43	16.62	**0.00**	**0.00**	1.89
TR	0.7	8	1.32	1.60	1.82	2.85	7.31	**0.00**	0.69	1.00	**0.01**	0.06	1.10	1.84	10.96	**0.00**	0.38	1.15
AP	0.75	4	0.31	0.31	1.92	5.61	27.52	**0.00**	0.38	1.72	**0.00**	**0.00**	**0.00**	5.49	28.27	**0.00**	**0.00**	**0.00**
AP	0.75	6	0.25	0.32	0.36	8.87	24.23	**0.02**	1.32	2.43	**0.00**	0.30	1.13	5.71	35.29	**0.00**	0.37	1.66
AP	0.75	8	0.35	0.35	1.32	7.90	22.61	0.52	1.64	3.82	**0.00**	0.46	1.25	4.19	30.31	**0.00**	0.12	1.00
URAND1000	0.75	4	1.10	1.65	3.32	20.57	309.43	0.57	2.37	7.76	**0.00**	**0.02**	**0.03**	13.19	49.64	**0.00**	**0.02**	0.18
URAND1000	0.75	6	1.76	4.09	5.64	10.81	38.92	1.55	3.49	6.44	**0.03**	0.06	0.10	14.87	88.71	**0.00**	**0.04**	0.72
URAND1000	0.75	8	2.37	6.16	8.85	9.26	81.06	2.65	5.73	8.65	0.09	0.21	0.66	14.84	75.07	**0.00**	0.22	0.58
WORLDAP	0.75	4	0.75	3.12	4.74	40.21	446.57	1.26	4.80	8.64	**0.00**	**0.00**	**0.02**	28.64	125.72	**0.00**	**0.03**	2.84
WORLDAP	0.75	6	2.25	2.67	6.33	57.12	144.59	3.33	7.86	12.37	0.20	0.38	1.22	16.36	49.19	**0.00**	1.92	6.55
WORLDAP	0.75	8	2.91	6.89	13.59	43.99	244.98	8.95	15.19	29.91	**0.00**	3.35	7.70	21.81	69.11	1.19	2.87	4.78
RAND5000	0.75	4	1.89	2.75	4.63	65.59	444.97	1.66	2.74	5.02	**0.00**	**0.05**	0.38	21.20	80.04	**0.00**	0.10	0.37
RAND5000	0.75	6	2.71	4.61	6.84	72.00	361.07	4.18	7.83	12.47	**0.00**	0.64	1.22	33.70	153.49	0.28	0.86	1.70
RAND5000	0.75	8	6.06	7.61	12.99	44.98	297.32	8.32	10.68	15.76	0.22	1.17	4.52	16.41	82.21	**0.00**	1.22	2.73

Table 7: The minimum gaps, median gaps and maximum gaps of solutions (in %) obtained by EHLC_GA/GVNS and NC_GA/GVNS as well as the median values and the maximum values of the median speedups (MSU) between EHLC and NC methods for USApHMPI. The numbers of instances that $Median\_Gap_{EHLC\_GA} \leq Median\_Gap_{NC\_GA}$ and $Median\_Gap_{EHLC\_GVNS} \leq Median\_Gap_{NC\_GVNS}$ are 15 and 17 out of 21, respectively. The gaps that are smaller than or equal to 0.05% are highlighted in bold.

USApHMPI			**EHLC_GA Gap**			**EHLC_GA MSU**		**NC_GA Gap**			**EHLC_GVNS Gap**			**EHLC_GVNS MSU**		**NC_GVNS Gap**		
Dataset	α	p	Min	Med	Max	Med	Max	Min	Med	Max	Min	Med	Max	Med	Max	Min	Med	Max
TR	0.3	4,5	0.07	0.07	3.93	1.69	5.53	**0.04**	0.51	2.52	**0.00**	**0.00**	**0.00**	2.60	10.55	**0.00**	**0.00**	**0.00**
TR	0.3	6,8	0.20	0.83	2.97	2.90	6.14	0.42	2.02	4.78	**0.00**	**0.00**	0.25	1.41	4.66	**0.00**	**0.00**	0.25
TR	0.3	8,12	2.00	3.11	3.61	2.48	11.46	1.00	2.31	6.80	**0.00**	0.46	1.18	1.79	5.34	0.08	0.84	1.18
TR	0.5	4,5	0.62	0.64	2.30	1.07	6.87	0.09	0.64	1.18	**0.00**	**0.04**	0.10	2.29	12.48	**0.04**	0.10	0.74
TR	0.5	6,8	1.96	2.95	5.27	1.85	3.85	0.45	1.95	3.41	0.34	0.51	1.66	1.57	4.52	**0.00**	0.56	0.77
TR	0.5	8,12	2.00	3.20	4.77	1.54	3.40	0.85	3.22	5.05	**0.00**	0.52	2.08	1.72	4.75	0.07	0.21	0.66
TR	0.7	4,5	1.82	1.82	2.36	2.02	7.23	0.63	0.79	1.74	**0.00**	**0.00**	0.59	1.78	6.91	**0.00**	**0.00**	0.59
TR	0.7	6,8	1.73	2.82	3.94	1.73	3.98	0.15	1.77	4.53	**0.00**	0.68	2.02	1.71	7.88	**0.00**	0.54	1.45
TR	0.7	8,12	2.03	3.31	4.72	2.05	3.27	0.82	2.88	5.69	**0.00**	0.49	1.95	1.37	3.33	**0.00**	0.53	4.20
AP	0.75	4,5	0.64	0.83	0.96	4.70	27.34	0.11	0.46	5.47	**0.00**	**0.00**	**0.00**	13.77	85.39	**0.00**	**0.00**	**0.00**
AP	0.75	6,8	0.29	1.51	3.75	3.11	8.52	0.88	2.29	5.00	**0.00**	0.15	1.17	4.09	40.71	**0.00**	0.14	0.94
AP	0.75	8,12	0.36	2.16	4.94	4.08	24.33	2.01	5.66	9.18	**0.00**	0.17	1.84	2.31	18.70	**0.00**	0.24	1.84
URAND1000	0.75	4,5	0.48	1.33	2.10	19.91	65.83	0.80	1.40	2.83	**0.00**	**0.00**	**0.00**	31.03	85.55	**0.00**	**0.00**	**0.01**
URAND1000	0.75	6,8	2.39	3.56	5.09	18.34	43.91	2.79	3.81	4.82	**0.00**	0.09	0.59	29.72	162.90	**0.00**	0.07	0.76
URAND1000	0.75	8,12	3.64	7.51	10.77	11.02	75.04	5.53	7.84	10.20	**0.03**	0.13	1.28	20.30	142.13	**0.00**	0.28	2.63
WORLDAP	0.75	4,5	0.69	1.48	6.86	27.05	256.05	0.80	3.81	6.06	**0.00**	**0.01**	0.30	41.21	371.62	**0.00**	0.06	2.94
WORLDAP	0.75	6,8	4.10	5.72	13.11	31.59	120.89	3.89	6.86	10.66	**0.00**	0.39	4.70	27.08	311.33	0.10	0.58	7.83
WORLDAP	0.75	8,12	8.77	12.25	16.24	36.79	126.83	6.62	13.13	19.58	**0.00**	3.32	6.24	24.25	221.49	0.90	7.08	15.11
RAND5000	0.75	4,5	0.60	1.55	2.26	47.85	534.43	1.25	1.68	3.09	**0.00**	**0.03**	**0.04**	46.46	188.00	**0.02**	**0.03**	0.18
RAND5000	0.75	6,8	3.00	4.81	6.27	49.07	192.76	3.59	4.90	9.03	**0.00**	0.28	0.44	41.31	116.61	**0.02**	0.48	0.83
RAND5000	0.75	8,12	4.93	8.09	9.13	44.75	186.24	7.51	8.91	12.24	**0.00**	0.11	1.56	38.23	144.30	**0.02**	0.84	1.63

Table 8: The gaps of solutions obtained by EHLC_Benders decomposition and NC_Benders decomposition and the median speedup (MSU) between EHLC and NC method for UMApHMPC.

UMApHMPC			**EHLC_Benders**	**EHLC_Benders**	**NC_Benders**
Dataset	α	p	Gap (%)	MSU	Gap (%)
TR	0.3	4	**0.00**	61.93	**0.00**
TR	0.3	6	**0.00**	74.78	**0.00**
TR	0.3	8	**0.00**	62.26	**0.00**
TR	0.5	4	**0.00**	37.97	**0.00**
TR	0.5	6	**0.00**	56.81	**0.00**
TR	0.5	8	**0.00**	61.28	**0.00**
TR	0.7	4	**0.00**	38.22	**0.00**
TR	0.7	6	**0.00**	49.78	**0.00**
TR	0.7	8	**0.00**	59.5	**0.00**
AP	0.75	4	**0.00**		17.07
AP	0.75	6	**0.00**		41.71
AP	0.75	8	**0.00**		38.35

Table 9: The gaps of solutions obtained by EHLC_Benders decomposition and NC_Benders decomposition and the median speedup (MSU) between EHLC and NC method for CMApHMPC.

CMApHMPC			**EHLC_Benders**	**EHLC_Benders**	**NC_Benders**
Dataset	α	p	Gap (%)	MSU	Gap (%)
TR40	0.3	4	**0.00**	23.25	**0.00**
TR40	0.3	6	**0.00**	18.43	**0.00**
TR40	0.3	8	**0.00**	31.03	**0.00**
TR40	0.5	4	**0.00**	13.14	**0.00**
TR40	0.5	6	**0.00**	17.45	**0.00**
TR40	0.5	8	**0.00**	24.04	0.07
TR40	0.7	4	**0.00**	20.66	**0.00**
TR40	0.7	6	**0.00**	27.37	0.26
TR40	0.7	8	**0.04**	23.96	**0.00**

Table 10: The gaps of solutions obtained by EHLC_Benders decomposition and NC_Benders decomposition and the median speedup (MSU) between EHLC and NC method for UMApHMPI.

UMApHMPI			**EHLC_Benders**	**EHLC_Benders**	**NC_Benders**
Dataset	α	p	Gap (%)	MSU	Gap (%)
TR40	0.3	4,5	**0.00**	28.18	**0.00**
TR40	0.3	6,8	**0.00**	0.95	**0.00**
TR40	0.3	8,12	**0.00**	29.37	**0.00**
TR40	0.5	4,5	**0.00**	40.2	**0.00**
TR40	0.5	6,8	**0.00**	38.46	**0.00**
TR40	0.5	8,12	**0.00**	28.97	**0.00**
TR40	0.7	4,5	**0.00**	33.67	**0.00**
TR40	0.7	6,8	**0.00**	34.22	**0.00**
TR40	0.7	8,12	**0.00**	38.9	**0.00**

6.4. Additional results for initial gap versus speedup.

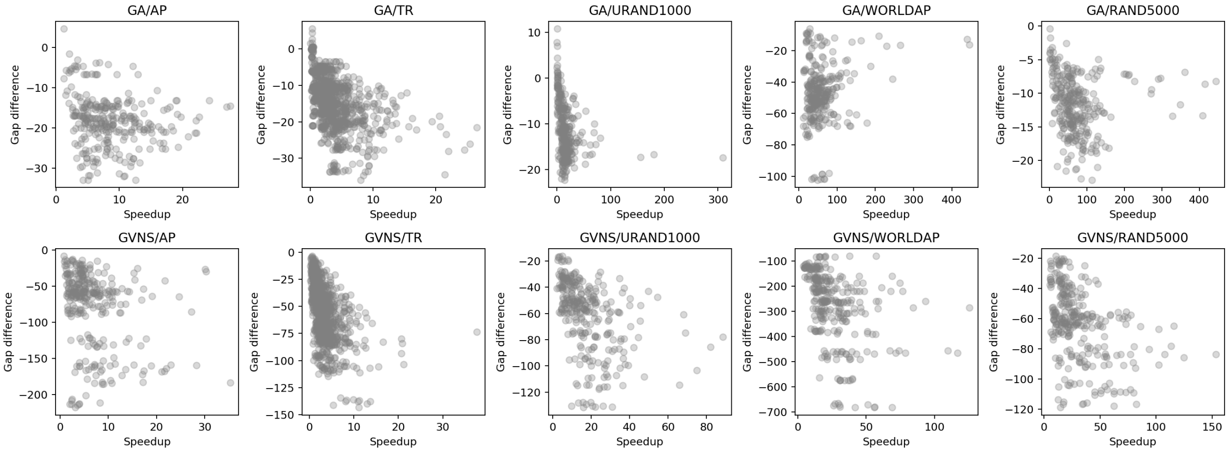


Figure 12: Correlation between gap difference and the speedup compared to non-contraction methods for CSApHMPC. The gap difference is measured as the initial gap of EHLC (after rewriting step) minus the gap obtained by non-contraction methods. Smaller initial gaps by EHLC compared to non-contraction methods lead to larger speedups.

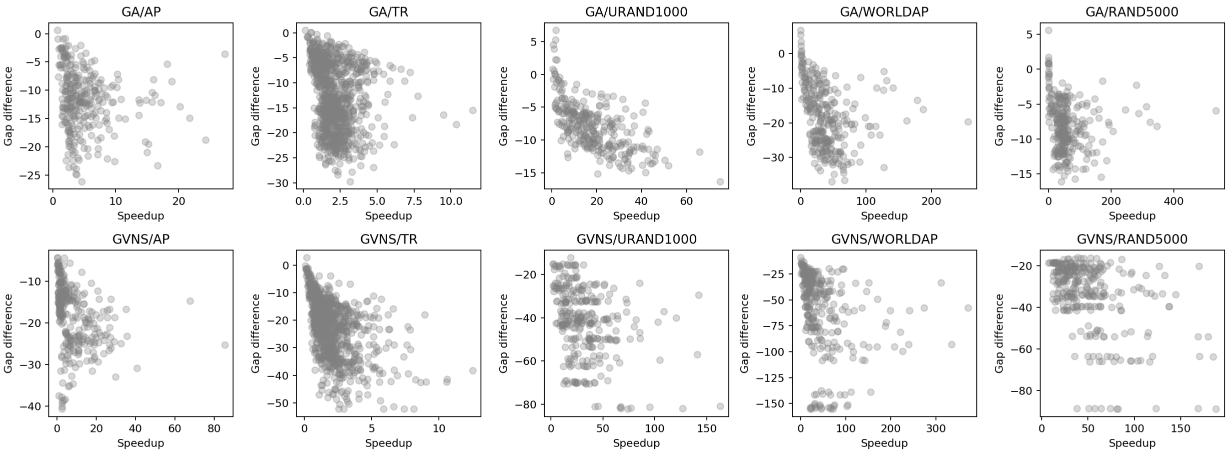


Figure 13: Correlation between gap difference and the speedup compared to non-contraction methods for USApHMPI. The gap difference is measured as the initial gap of EHLC (after rewriting step) minus the gap obtained by non-contraction methods. Smaller initial gaps by EHLC compared to non-contraction methods lead to larger speedups.