Journal of Project Management 9 (2024) 277–300

Contents lists available at GrowingScience

Journal of Project Management

homepage: www.GrowingScience.com

Optimizing Modular Hub Location in Air and Road Transportation Systems

Zahra Arabzadeh Nosrat Abad[a] and Farid Momayezi[a*]

[a]*Department of Industrial Engineering, Urmia University of Technology, Iran*

C H R O N I C L E

Article history:
Received: January 20, 2024
Received in revised format: March 1, 2024
Accepted: March 16, 2024
Available online: March 16, 2024

Keywords:
Hub Network
Modular Hub Location Problem (MHLP)
Mixed-Integer Programming (MIP)
LP Relaxation-based Method
Genetic Algorithm (GA)
Taguchi method

A B S T R A C T

Hub networks play a crucial role in optimizing transportation costs in air and road systems. Their main objective is to strategically locate hubs and allocate non-hub nodes within the network. The modular hub location problem is a specific area of hub network design that focuses on accurately calculating transportation costs, considering factors like trip numbers and capacity constraints in network routes. This study proposes a mixed-integer programming model to address the modular hub location problem with multiple allocations. It considers dependent and independent costs associated with vehicles per trip between hub network routes, considering specific vehicle capacities. Two datasets are utilized for validation: the CAB dataset representing 25 nodes of US airports and the TR dataset representing the Turkish transportation system with 81 nodes. To tackle the NP-hard nature of hub location models and the computational complexity of the proposed model, two solutions are developed. Firstly, a novel LP relaxation-based method using GAMS software provides near-optimal solutions for medium-sized instances. Additionally, a Genetic Algorithm (GA) implemented in MATLAB handles larger instances. The GA's efficiency is enhanced by tuning its parameters using the Taguchi method. Results analysis shows that both proposed algorithms yield high-quality solutions within significantly reduced timeframes compared to the CPLEX solver in GAMS software. The LP relaxation-based method performs well for medium-sized instances, while the GA approach is efficient for larger instances after parameter tuning with the Taguchi method.

1. Introduction

Hub location problems (HLPs) play a significant role in both air and road transportation systems. In traditional transportation networks, the flow of goods or passengers is typically directed straight from the source node to the destination node. However, due to resource limitations, it may not always be possible to establish direct transport routes between all nodes in the network. As a result, HLPs aim to optimize network transportation costs by selecting a subset of nodes called hubs. These hubs act as consolidation and redistribution points, facilitating indirect flow routing strategies (Alumur et al., 2016). The main objective of HLPs is to strategically choose the hub nodes, allocate non-hub nodes to the hubs, and determine the most efficient routes for transmitting flow between origin and destination nodes within the network. HLPs encompass various types, including hub location, p-hub median, hub covering, and p-hub center problems. These types can be extended based on factors such as non-hub node allocation patterns to hubs. This allocation can be approached through two primary methods: single-allocation and multiple-allocation. In the single-allocation approach, each non-hub node is assigned to a single hub, while in the multiple-allocation approach, non-hub nodes can be allocated to multiple hubs. These allocation strategies have significant implications for the efficiency and performance of the hub network (Alumur et al. 2016). The other important factor in the classification of HLP is the consideration of capacity constraints in hubs. This factor distinguishes HLP types into capacitated and uncapacitated forms.

* Corresponding author.
E-mail address: farid.momayezi@uut.ac.ir (F. Momayezi)

doi: 10.5267/j.jpm.2024.3.001

Additionally, Farahani et al. (2013) introduced five additional scenarios for the HLP classification based on the objective function and allocation strategy. Based on Farahani et al., the HLP can have the following characteristics:

- **Problem Domain:** The problem domain in hub location studies can be classified into three types: network-based, discrete, and continuous. In the network-based domain, all nodes in the network have the potential to serve as hub locations. In the discrete domain, there is a predefined and specific set of nodes that are designated as hubs. The main focus of this problem is to determine the optimal locations for these hubs within the network. On the other hand, in the continuous domain, the problem allows for hubs to be located at any point in a two-dimensional plane or three-dimensional space. This means that the hub locations are not limited to specific predefined nodes, and the problem involves finding the optimal locations for the hubs within the continuous space.
- **Objective Function:** In HLP, there are three types of objective functions commonly used: minimum-maximum, minimum-sum, and maximum-sum.

A. The minimum-maximum objective function minimizes the highest transportation costs within the network, reducing the impact of expensive routes.
B. The minimum-sum objective function minimizes the overall total transportation costs, considering the collective sum of costs across the network.
C. The maximum-sum objective function maximizes the total flow covered by selected hub locations, optimizing flow capacity utilization.

- **Determining the Number of Hubs**: In HLP, the approaches can be categorized as exogenous or endogenous.

A. Exogenous approach: The number of hubs is predetermined or predefined before solving the problem. The specific number of hubs to be included in the network is already known or set.
B. Endogenous approach: The number of hubs is not predefined. Instead, it is determined as part of the problem-solving process. The optimization algorithm or model itself determines the suitable number of hubs based on given constraints and objectives.

- **Hub Capacities:** Hubs can have either limited capacities or unlimited capacities in the network.
- **Allocation of Non-Hub Nodes to Hubs**: There are two types of allocation strategies for hub networks: single-allocation and multiple-allocation. Single-allocation assigns each non-hub node to a single hub, while multiple-allocation allows non-hub nodes to be assigned to multiple hubs.

These characteristics define different variations and configurations of the HLP, allowing for various modeling and solution approaches. The HLP assumes that flows in the network are transported between origins and destinations through a set of hubs. The network includes connections between non-hubs to hubs, hubs to hubs, and hubs to non-hubs. In traditional HLP studies, transportation costs on hub-to-hub paths are often assumed to have a fixed percentage discount, regardless of the flow volume. However, this assumption can lead to inaccurate overall transportation cost calculations and suboptimal hub locations. To address these issues, researchers have introduced flow-dependent cost-based networks called the Modular Hub Location Problem (MHLP), where transportation costs are modularly dependent on flow volumes (Farahani et al., 2013; Hosseini, 2013). Modular transportation costs are highly relevant in various domains, such as freight transportation networks. In the MHLP, transportation involves a fixed number of vehicles with specific capacities matching the required capacity for each path in the hub network. Accurate estimation of transportation costs in modular hub networks considers factors like the number of vehicles, their capacities, and the distances covered. This precise estimation enhances decision-making in hub network design, leading to more effective outcomes. The MHLP aims to minimize the overall cost of the network, considering various components such as hub installation, transportation, and vehicle costs. It incorporates capacity factors into the network routes. Research by Tanash et al. (2017) supports the practicality and accuracy of the MHLP model, highlighting its superiority over classical HLP models in capacity-constrained scenarios. The objective of the present study is to propose a novel mixed-integer programming (MIP) model for the MHLP with multiple allocations. This model considers optimal fixed capacities for each vehicle on each route in the network, depending on the vehicle type. For example, smaller- capacity airplanes or trucks may be assigned to routes between non-hub-to-hub nodes and hub-to-non-hub nodes, while larger capacity airplanes or trucks may be used for routes between two hubs. In multiple allocation, there may exist multiple routes between a source and destination node, and the flow is distributed among these routes due to limited transport vehicle capacity and cost reduction in the network. The research introduces a continuous variable for multiple allocations, representing the flow percentage sent through each route. This approach improves flow distribution and network performance. The proposed MIP model optimizes flow distribution, considering cost and capacity limitations, resulting in balanced and efficient flow management within the network. In addition, the present study considers the cost of using transportation vehicles (such as airplanes or trucks) for each trip in the objective function to calculate the overall network cost. This approach leads to a more accurate determination of the number of flights or trucks required, depending on whether air or road transportation is being used, with the objective of minimizing the overall cost.

The proposed model does not consider a discount coefficient for goods between hubs and also coefficient factors for collecting and distributing goods within the hub network because the entire cost of using a transportation vehicle is incurred regardless of whether the vehicle is operating at its maximum capacity or not. Flow transmission costs in the network are considered dependent (distance-related) and independent (fixed) costs, including fuel consumption, wages, maintenance, and repair expenses for each trip. Finally, the proposed model will be solved using the CPLEX solver embedded in GAMS software, and due to the complexity of the solution space of the model to achieve the best possible solution, an innovative approach based on linear relaxation of integer variables in GAMS and a metaheuristic algorithm based on a genetic algorithm (GA) will be employed in MATLAB software. To enhance the efficiency of the proposed algorithm, the GA parameters will be tuned using the Taguchi method. The main objectives of the presented study can be summarized as follows:

- Present a novel model for the modular hub location problem with multiple allocations.
- Calculate the precise transportation cost in the hub network.
- Utilize the capacity of vehicles instead of discount factors.
- Determine the capacity of the vehicle in the hub network.
- Determine the number of flights or trucks transported on each path in the hub network.
- Employ a continuous variable between zero and one instead of a binary variable to allocate non-hub nodes to hubs, enabling the calculation of the percentage of flow that can be sent through each path in the network.
- Develop a new LP relaxation-based method to reduce runtime for medium-sized samples.
- Devise a GA to solve the presented problem for large-sized samples.
- Enhancing the efficiency of the GA through the tuning of GA parameters.

The rest of the paper is structured as follows:

Section 2 provides a literature review; Section 3 describes the problem and formulates a mathematical model; Section 4 presents solution algorithms; Section 5 showcases results and examples; and Section 6 summarizes the main conclusions.

2. Literature review

2.1. Classical Hub Location Problem

The field of HLP has received significant attention from researchers, leading to a substantial body of literature over the years. Several comprehensive review papers have been published in this field. Some noteworthy references include Alumur and Kara (2008), Campbell and O'Kelly (2012), Zanjirani Farahani et al. (2013), Contreras and O'Kelly (2019), and Alumur et al. (2020). These papers provide valuable insights into the HLP literature and serve as important resources for understanding the advancements in this area. The HLP was formulated by O'Kelly (1987) for the first time. The formulation presented by O'Kelly is a non-linear type of model. Campbell (1994) presented the integer linear programming models for this type of problem. Ernst and Krishnamoorthy (1996, 1999) developed practical models using flow-based network modeling. This network-based modeling approach, compared to the model proposed by Skorin-Kapov et al. (1996), has the advantage of fewer variables. The first concept of modular capacity in HLP was introduced by Jaillet et al. (1996), but they did not directly refer to the means of modularity in their study. Their study was in the air transportation system and designed flow-based models for capacitated networks and routing policies, which means the concept of modular. Subsequently, the superiority of flow-based models over path-based models was demonstrated by O'Kelly (1998) and Bryan & O'Kelly (1999). Furthermore, the first series of these models were studied by O'Kelly & Herner (2001), where a non-linear cost function based on the connectivity performance function was considered, and all links were economically connected to each other. The papers with flow-based models pointed out that the discount factor that has been considered between hub nodes should be regarded as a function of the flow volume. Carello et al. (2004) extended the hub location problem with capacity for selecting hub nodes in a telecommunications network using a local search algorithm. The HLP is widely recognized as an NP-hard problem, primarily due to its computational complexity, which persists even for small problem sizes. As the size of the problem increases, the computational time required to solve it grows significantly. This complexity stems from the combinatorial nature of the problem, where the number of possible solutions increases exponentially with the size of the problem. Consequently, finding an optimal or near-optimal solution for the HLP can be time-consuming, posing a challenge even for small instances of the problem. As a result, researchers have focused on proposing new models to address specific variations of the HLP and have presented solution methods to effectively solve their proposed models. To investigate the various models presented for HLPs and methods proposed for solving HLPs, researchers have conducted extensive review studies on HLP.

2.2. Modular Hub Location Problem

The concept of modularity in the Hub Location Problem (HLP), which refers to the capacity and packing of flow between network links, was initially introduced by Yaman (2005). This type of model is typically categorized as a flow-based model due to its emphasis on optimizing flow allocation and management within the network. The concept was further expanded upon by Yaman and Carlo (2005), who formulated the problem as a quadratic mixed-integer programming problem. In their formulation, the cost functions for network links were made dependent on the flow. Yaman (2008) addressed the HLP with modular arc capacity, proposing two formulations and a heuristic algorithm based on Lagrangian relaxation and local search. The study aimed to minimize link installation costs by selecting P hubs from a given set of nodes, with each non-hub node connected to one hub and each hub connected to a central hub. Mirzaghafour (2013) introduced the main type of modular hub location problem, known as the MHLP model, which uses the number of facilities on paths of the network instead of the cost per unit flow. The study incorporated the concept of modular capacity for vehicles used in the hub network and presented a new formulation. Alumur et al. (2016) tackled the HLP with capacity constraints, specifically focusing on the single assignment version. It introduces a realistic variant where edge capacities between hubs are increased modularly. The study proposed efficient heuristic methods that outperform previous approaches, utilize memory structures to enhance search algorithms, and are compared with previous heuristics using benchmark instances and incorporating frequency information in the constructive method. Corberán et al. (2016) addressed the capacitated single-assignment HLP with modular link capacities. They developed a metaheuristic algorithm based on strategic oscillation, originally used in tabu search. The algorithm incorporated constructive and destructive algorithms, along with local search procedures, to balance diversification and intensification for efficient search. Computational results on a large set of instances demonstrate that the proposed metaheuristic finds high-quality solutions quickly, outperforming previous tabu search implementations. Tanash et al. (2017) conducted a study on the MHLP in hub-and-spoke networks. Their research aimed to reduce transportation costs by consolidating flows at hub facilities. To address this problem, they proposed a branch-and-bound algorithm that utilized a Lagrangean relaxation technique to obtain both lower and upper bounds. The study also presented numerical results for benchmark instances featuring up to 75 nodes. Hoff et al. (2017) addressed the HLP, minimizing transportation costs while meeting capacity constraints. They proposed efficient heuristic methods that outperformed previous approaches, integrating modular capacity increases and incorporating frequency information for traffic routing. Bashiri et al. (2018a) investigated the integration of manufacturing processes with transportation, emphasizing the utilization of manufacturing hubs to decrease transportation and manufacturing costs. The study focused on manufacturing hubs equipped with automation modules, where the hub capacity depended on the allocated modules. To accommodate changing demands, the paper proposed transferring movable automation modules among hubs. A mathematical model was developed to optimize hub locations, automation module allocations, and module transfers across different time periods. Correia et al. (2018) presented multi-period stochastic capacitated multiple allocation hub location problems that assume uncertain demand with modular capacities in hubs. Karimi and Bashiri (2019) examined the MHLP, considering concepts such as commodity split, multimodal transportation systems, and capacity modulation. Their study specifically focused on the added complexity arising from flow division.

Mikic´ et al. (2019) presented the capacitated modular hub location problem (CMHLP) with single allocation conditions. They considered capacity on hubs and paths of the network and presented a general variable neighborhood search to solve their proposed model. Keshvari Fard and Alfandari (2019) focused on the MHLP, where transportation costs are modeled as a stepwise function rather than linear. The study explored formulations solved by exact methods and proposed a specific generalized linear cost function that closely approximates the stepwise cost function. Additionally, the paper examined the cost savings associated with direct shipments in hub and spoke networks, leveraging the stepwise cost function's ability to incorporate direct transportation. Momayezi et al. (2021) developed a MILP model for designing modular hub networks under disruptive conditions, utilizing an adaptive large neighborhood search algorithm for solving problems of larger size. Wu et al. (2023) addressed the CMHLP in the design of telecommunications networks. They proposed a parallel adaptive memory algorithm that stores solution blocks and complete solutions in a shared memory for creating offspring. The algorithm incorporates recombination and mutation operators for search diversification, a tabu search procedure for search intensification, and parallel computing for global optimization. Computational results on benchmark instances showed that their algorithm outperformed existing heuristics, achieving improved best-known solutions in over 67% of the cases. AL Athamneh et al. (2023) addressed the Incomplete Hub Location Problem with Modular Capacity and Direct Connections (MHLPDC) in distribution systems. The MHLPDC aims to minimize network costs by locating hub facilities, connecting non-hub nodes to hubs, and activating various links. The study presents a mixed-integer mathematical programming formulation and a heuristic algorithm based on a greedy randomized adaptive search and variable neighborhood search approach. Computational experiments on benchmark instances, ranging up to 40 nodes, demonstrate the algorithm's capability to generate high-quality solutions. Additionally, a sensitivity analysis of the optimal network structure reveals the impact of varying hub and access arc capacities, variable costs, and the discount factor on hub establishment and direct shipments. Khaleghi and Eydi (2023) presented a bi-objective, nonlinear mathematical model for designing a multi-period hub network. They considered the capacity of the hub in their proposed model to be modular. Each module corresponds to a number of servers at hubs.

2.3. The Genetic Algorithms for Solving Various Hub Location Problems

The HLP is a known NP-hard problem, and the Genetic Algorithm (GA) has emerged as one of the most effective methods for addressing this challenge. Notably, Abdinnour-Helm and Venkataramanan (1998) were the pioneers in applying the GA to solve the HLP, paving the way for its utilization in this domain. Building upon this foundation, Topcuoglu (2005) extended the application of GA to solve the uncapacitated HLP by incorporating Tabu Search, a complementary metaheuristic technique. This combination of GA and Tabu Search demonstrated promising results in tackling the uncapacitated HLP. These studies underscore the practicality and efficacy of employing the GA as a valuable approach to solving the HLP and its variations. Damgacioglu (2015) presented GA to solve an uncapacitated single allocation planar hub location problem (PHLP), which is a type of HLP. Peker et al. (2016) explored the impact of delivery service requirements on transportation system configuration using a bi-objective single allocation p-hub center-median problem (BSpHCMP) and proposed an evolutionary algorithm that outperforms the non-dominated sorting genetic algorithm (NSGA-II) and PAES algorithms in solving the problem, with experimental results validating the proposed model and solution approaches. Majima et al. (2017) addressed determining the optimal location for hub stations in scheduled transportation services across various modes of transportation. The proposed method combines a growing network model for public transit network generation with a GA and cuckoo search algorithm for the HLP. The paper presented a comparison of the GA and cuckoo search algorithms for the HLP within this framework. Bashiri et al. (2018b) used GA, along with tuned parameters and a simulated annealing algorithm, to solve the mobile p-hub location problem in a dynamic environment. Rashidi Kahag et al. (2019) addressed a multi-objective intermodal hub-location-allocation problem using a GA. The problem involved minimizing total costs and system time, considering queuing systems at origin and destination hubs. A validated constrained bi-objective optimization model was developed, and the proposed GA outperforms other algorithms, as demonstrated by multi-objective metrics and the entropy-TOPSIS method. Basirati et al. (2020) addressed a many-to-many hub location-routing problem using a multi-objective imperialist competitive algorithm (MOICA). The proposed model minimizes total costs and achieves superior performance compared to NSGA-II, particularly for large-scale instances.

Rabbani et al. (2021) used a hybrid of k-means and genetic algorithms (KGA) to solve the p-Hub location-allocation problem and compared it to constraint programming (CP) methods. The CP model is better for memory usage but slower, while the KGA is faster but requires more memory. KGA also uses a new adaptive crossover operator for better solutions. In the study conducted by Mrabtiv (2022), a genetic algorithm was proposed and utilized in mathematical models to compare the performance of collaborative distribution hub network design scenarios. These scenarios were assessed based on sustainability indicators. The genetic algorithm was employed for both small and large instances, providing an effective approach for optimizing the network design. In the study conducted by Bhattacharjee and Mukhopadhyay (2023), two meta-heuristic algorithms, GA and its refined version, were proposed and compared for the Hierarchical Single-Allocation Hub Median Facility Location Problem (SA-H-MP). The performances of these algorithms were evaluated against CPLEX (exact method), simulated annealing (SA), and iterated local search (ILS). The refined GA demonstrated superior results compared to other meta-heuristics in more than 88% of the cases when validated on the CAB benchmark dataset. De Freitas et al. (2023) presented a solution for the two-level hub location routing problem with directed tours using the biased random-key genetic algorithm (BRKGA) metaheuristic. The BRKGA, calibrated with machine learning techniques, demonstrates superior performance compared to existing literature, producing high-quality solutions for instances with unknown optimal values. Wang et al. (2023) investigate a hub location-routing problem with third-party logistics (3PL) and propose a two-stage MIP model. They address uncertainty using an adaptive, distributionally robust model and apply a GA with local search for large-scale problem solving.

Based on the analysis of studies conducted in the field of the Modular Hub Location Problem, it is evident that all proposed models require a significant amount of time for problem-solving. As a result, multiple solutions have been introduced in these studies to address this issue. These advancements in algorithmic approaches make substantial contributions to the field of MHLP, providing more effective and efficient solutions for large-scale samples.

This study presents a novel MHLP model that aims to calculate network costs based on real constraints and costs, such as transportation costs and the cost of using vehicles per trip. Previous studies have primarily focused on either a developed exact method or a metaheuristic method to solve their presented model. However, to address the time complexity, we integrate both an exact method, the LP relaxation-based method, and a metaheuristic method, GA, into our proposed framework. It is worth highlighting that while various types of genetic algorithms have been developed for different HLPs, their application specifically for solving MHLPs has been largely unexplored. Therefore, this study not only introduces GA as a solution approach for MHLP but also investigates its effectiveness and efficiency in comparison to other methods. Additionally, we incorporate a tuning parameter for GA using the Taguchi method, a novel approach that has not been previously explored in the context of MHLP. By comparing the solutions obtained from both the LP relaxation-based method and GA, we evaluate their performance and provide insights into the applicability of these approaches for solving the presented MHLP model.

3. Problem Definition and Formulation

3.1. Problem Definition

One of the critical challenges faced by companies whose core business involves collection, distribution, and transportation, such as postal services, is the reduction of transportation costs, which constitute a significant portion of their overall expenses. Consequently, optimizing the calculation of trip quantities on each route within a mathematical model, employing a minimization objective function, proves invaluable for logistics firms in making informed decisions about procuring or leasing the appropriate number of vehicles with specific capacities. This research endeavors to reevaluate the computations of the HLP by focusing on the calculation of vehicle trip quantities. By conducting sensitivity analyses on vehicle capacities, it becomes possible to make well-informed decisions regarding the implementation of hub-based transportation networks. Since the flows are wrapped within the vehicles, their capacity serves as the determining factor for transportation. This research considers the capacity of vehicles between hub-to-hub transportation to be larger than that of vehicles between non-hub-to-hub and hub-to-non-hub transportation. Fig. 1 and Fig. 2 depict a network where the circles represent the origins and destinations, and the squares represent the hubs of the network in air and road transportation.

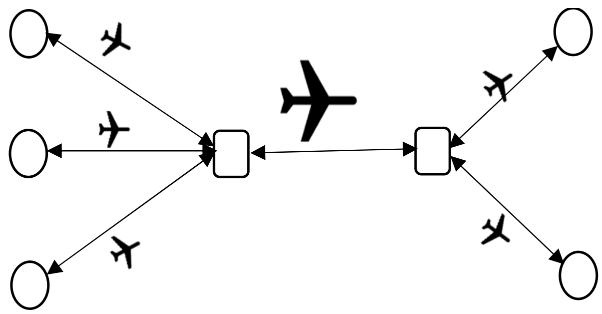

Fig. 1. Part of air transportation in a hub network with airplanes following designated paths

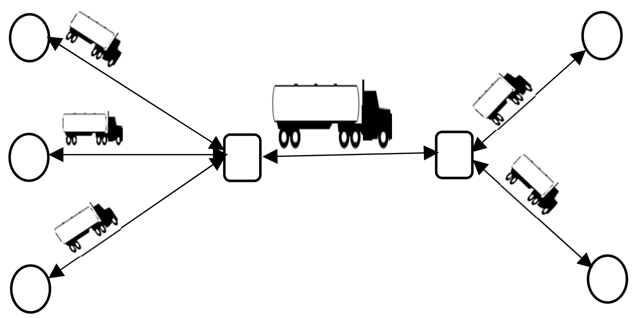

Fig. 2. Part of road transportation in hub network with trucks following designated paths

The hub network concept considers an aggregation factor for gathering flows from origin to hub, a discount factor α to account for flows between hubs, and a distribution factor for flows that are transmitted from hub to destination in the network. However, in the presented model, the vehicle capacity is utilized instead of these factors. Fig. 3 demonstrates the coefficient factors in the paths of the hub network.

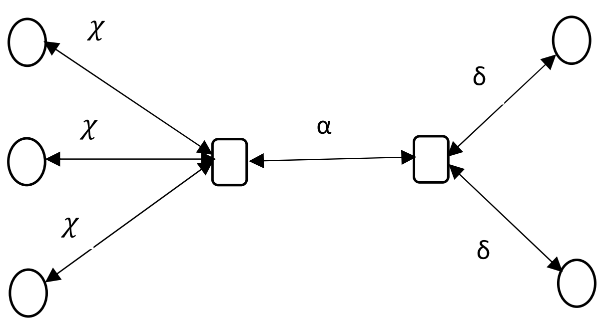

Fig. 3. Coefficient factors in the hub network

The datasets mentioned in the literature demonstrate variations in the values of χ, α, and δ. In particular, when considering the CAB dataset, both χ and δ are assigned a value of 1. Additionally, the value of α within this dataset ranges between 0 and 1, indicating a proportion of the volume of flows transmitted between hubs. It's worth noting that for the other datasets, although the values of χ and δ are greater than 1, the value of α still follows the same range of 0 to 1, similar to the CAB dataset. In practical scenarios, flows are transmitted using vehicles within the network, and each vehicle has a limited capacity. Increasing the volume of commodities and passengers sent will result in a higher discount, but both commodities and passengers are constrained by the vehicle capacity and cannot exceed it. If a vehicle's capacity is fully utilized and there is still additional flow to be sent along the path, another vehicle will be dispatched on that route. In such cases, the cost of both vehicle transmissions will be incurred, even if the capacity of the second vehicle is not fully utilized. Therefore, the coefficient of gathering and distribution factors of the flow will be calculated automatically between hub-to-non-hub and non-hub-to-hub routes.

The hub network operates by consolidating flows at hubs and distributing them to other hubs. To facilitate this process, larger-capacity vehicles are utilized, exceeding the capacity of vehicles used for transportation between other nodes and hubs. Notably, the cost per unit of goods or passengers transported by larger-capacity vehicles is lower compared to smaller ones, considering the proportional utilization of their capacity. As a result, an automatic discount factor coefficient is applied to appropriately adjust for this cost disparity during the optimization process. In the present research, the focus is on considering the number of vehicles with the specific capacity rather than incorporating coefficient factors between network paths. The proposed model adopts a multiple-allocation approach. The aim of implementing multiple allocations is to enhance reliability and achieve a balanced distribution across the network by utilizing multiple routes between origins and destinations. This approach allows for the existence of more than one route through hubs when transporting goods from a source to a destination. In this study, continuous variables are employed instead of binary variables for allocation within the proposed model. The utilization of continuous variables, ranging between zero and one, facilitates the calculation of the proportion of flow allocated to each route within the hub network. Furthermore, in the presented model, the number of flights of airplanes for air transportation or trips of trucks for road transportation is also calculated for each link within the hub network. This information provides insight into the frequency of transportation operations occurring along each specific link. To formally define the problem, consider a network G=(N, E), where N represents the set of nodes as demand points and potential hub locations, and E represents the set of links in the network connecting the nodes to each other. The symbols, parameters, and variables used in modeling the problem are presented in the table below:

Sets	
N	Set of nodes
E	Set of edges
Indices	
i, j	Origin and Destination nodes indices
k, m	Hub nodes indices
Parameters	
f_k	Fixed cost of hub installation at node k
FCR	Fixed cost per trip of vehicle (airplane/truck) with large capacity from hub to hub
FCQ	Fixed cost per trip of vehicle (airplane/truck) with small capacity from non-hub to hub
FCP	Fixed cost per trip of vehicle (airplane/truck) with small capacity from hub to non-hub
d_{ij}	Distance between node i and node j
w_{ij}	Flow between node i and j
Cap^{small}	Capacity of small vehicle (airplane/truck)
Cap^{large}	Capacity of large vehicle (airplane/truck)
Decision Variables	
$Z_k = \begin{cases} 1 \\ 0 \end{cases}$	If node k is selected as a hub Otherwise
X_{ijkm}=	Portion of flow w_{ij} sent from node i to node j via hubs k and m
P_{ik}=	Number of small capacity vehicle (airplane/truck) used between node i and hub k
R_{km}=	Number of large capacity vehicle (airplane/truck) used between hubs m and k
Q_{mj}=	Number of small capacity vehicle (airplane/truck) used between node j and hub m

Fig. 4 shows the use of P_{ik}, R_{km}, *and* Q_{mj} variables with their capacity in the routing of the hub network instead of using X_{ijkm} in the whole of the network calculation.

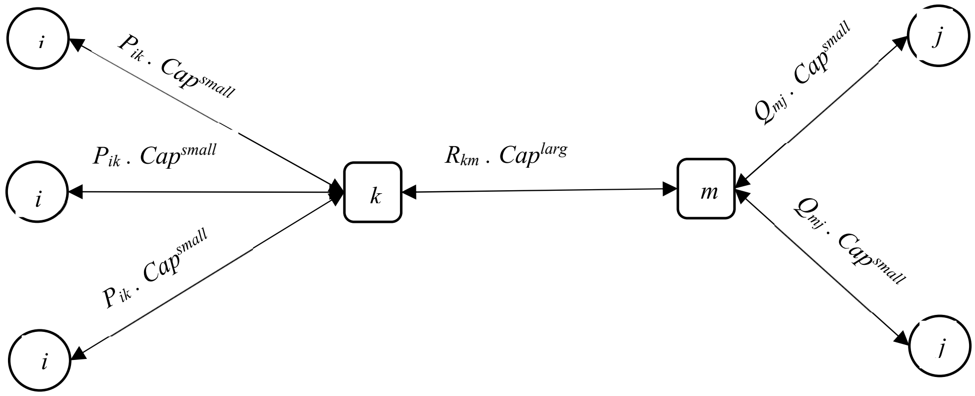

Fig. 4. Application of P_{ik}, R_{km}, *and* Q_{mj} with their capacity in the rout of the hub network

Furthermore, in the presented model, the capacity of each hub route in the network can be calculated as follows:

- For non-hub-to-hub routes, the capacity is determined by multiplying the number of vehicles (P_{ik}) with the capacity of smaller vehicles (Cap^{small}).
- For routes between hubs, the capacity is calculated by multiplying the number of vehicles (R_{km}) with the capacity of larger vehicles (Cap^{larg}).
- For hub-to-non-hub routes, the capacity is determined by multiplying the number of vehicles (Q_{mj}) with the capacity of smaller vehicles (Cap^{small}).

This approach allows for the determination of the required capacity for different types of routes within the hub network.

In the literature, the cost of flow transmission between nodes (C_{ij}) in the network is typically associated with the distance between nodes (d_{ij}) due to the dependence on fuel consumption. However, in this study, the cost of flow is further divided into two distinct components. The first component is the dependent cost, which remains consistent with the previous research and is directly proportional to the distance between nodes (d_{ij}). This component captures the fuel consumption and associated costs incurred during the transportation process.

The second component is the independent cost or fixed cost (*FCP and FCQ* for vehicles with small capacity and *FCR* for vehicles with larger capacity). This cost includes the maintenance cost of the vehicle and the wages of the vehicle crew per trip. Unlike the dependent cost, which varies with the distance between nodes, the independent cost remains relatively constant irrespective of the distance traveled.

3.2. Mathematical Model

A mixed-integer linear programming model is presented modular hub location problem with for multiple allocation as follow.

$$\min \sum_{k\in N} f_k Z_k + \sum_{i\in N}\sum_{k\in N} D_{ik} P_{ik} + \sum_{k\in N}\sum_{m\in N} D_{km} R_{km} + \sum_{m\in N}\sum_{j\in N} D_{mj} Q_{mj} + \sum_{i\in N}\sum_{\substack{k\in N\\ k\neq i}} FCP.P_{ik} + \sum_{k\in N}\sum_{\substack{m\in N\\ m\neq k}} FCR.R_{km} + \sum_{m\in N}\sum_{\substack{j\in N\\ j\neq m}} FCQ.Q_{mj} \tag{1}$$

subject to:

$$\sum_{k\in N}\sum_{m\in N} X_{ijkm} = 1 \qquad i\in N, j\in N \tag{2}$$

$$\sum_{m\in N} X_{ijkm} + \sum_{\substack{m\in N\\ m\neq k}} X_{ijmk} \le Z_k \qquad i\in N, j\in N, k\in N \tag{3}$$

$$\sum_m \sum_j w_{ij} X_{ijkm} \le P_{ik} Cap^{small} \qquad i \in N,\ k \in N \tag{4}$$

$$\sum_i \sum_j w_{ij} X_{ijkm} \le R_{km} Cap^{large} \qquad k \in N,\ m \in N \tag{5}$$

$$\sum_i \sum_k w_{ij} X_{ijkm} \le Q_{mj} Cap^{small} \qquad m \in N, j \in N \tag{6}$$

$$X_{ijkl} \ge 0 \qquad i \in N, j \in N,\ k \in N,\ m \in N \tag{7}$$

$$Z_k \in \{0,1\} \qquad k \in N \tag{8}$$

$$P_{ik}, R_{km}, Q_{mj} \ \text{positive integer} \qquad i \in N, j \in N,\ k \in N,\ m \in N \tag{9}$$

Expression (1) represents the objective function of the problem, which aims to minimize the overall cost. The first part of this objective function pertains to the cost of establishing hubs. The second part calculates the cost of sending flow between node **i** and hub **k** using vehicles (airplanes in air transportation or trucks in cargo) with small capacity, essentially capturing the transportation cost between origins and hubs. The third part considers the transportation cost between hubs with the large capacity of the vehicle (nodes **m** and **k**). The fourth part of the objective function calculates the cost of transportation from hub to non-hub (nodes **k** and **j**). The second to fourth parts of the objective function calculate the dependent cost of transportation. Each part is determined by multiplying the number of vehicles by the distance of the route. The fifth to seventh parts of the objective function calculate the independent cost of transportation. This cost is determined by considering the usage of vehicles with specific capacity for transporting flow between different points in the network.

The fifth part focuses on the fixed cost of using vehicles with a small capacity to transport flow from origins to hubs (between nodes i and k). This component captures the expenses associated with utilizing vehicles with small capacity for transporting flow from the starting points to the designated hubs. The sixth part calculates the independent cost of transportation for hub-to-hub routes (between nodes m and k). This component accounts for the costs incurred when using vehicles with large capacity to transport flow between different hubs within the network. The seventh part addresses the fixed cost of using vehicles with a small capacity to transport flow from hubs to destinations (between nodes m and j). This component captures the expenses involved in utilizing vehicles with small capacity for transporting flow from the hubs to the final destinations.

Constraint (2) ensures that the flow between source i and destination j is routed through hubs k and m. Constraint (3) states that if node i is assigned to hub k, then the flow from node i to node j is transferred through hub m. Constraint (4) ensures that the total flow sent from source i to hub k is less than or equal to the number of flights/trips conducted from node i to hub k (P_{ik}) multiplied by the vehicle capacity. Similarly, constraint (5) indicates that the total flow sent from hub k to hub m is less than or equal to the number of flights/trips conducted from hub k to hub m (R_{km}) multiplied by the vehicle capacity. Likewise, constraint (6) states that the total flow sent from hub m to destination node j is less than or equal to the number of flights/trips conducted from hub m to node j (Q_{mj}) multiplied by the vehicle capacity. Constraints (7) to (9) define the decision variable types in the model.

4. Solution algorithm

The purpose of conducting computational experiments is to evaluate the performance of the proposed model. The presented model is solved using three methods. Firstly, the model is solved in the GAMS software, version 25.1.2, which is implemented using the CPLEX optimization engine. It was executed on a computer with the following specifications: 16.00 GB of RAM and an Intel Core i7 CPU (2.8 GHz). However, since the presented model is time-consuming and requires a significant amount of time to obtain the best-known solution, even for small-sized datasets, two additional algorithms have been developed to address this issue in the GAMS software.

The second approach involves an exact solution based on the LP relaxation method, which is specifically designed to solve medium-sized problems more efficiently. This method is implemented within the GAMS software itself. The third approach utilizes GA, which has been developed in the MATLAB software and is specifically tailored to handle larger problem sizes. By utilizing these GA, it is possible to find solutions to larger instances of the problem within a reasonable time frame. By comparing and evaluating the performance of these different solving methods, it is possible to determine the strengths of the strengths of the proposed algorithms in terms of efficiency and solution quality.

4.1. LP Relaxation Base Method Algorithm

Given the computational challenges inherent in the proposed model involving 10, 15, 20, and 25 nodes in the CAB dataset, it has been observed that using the CPLEX solver in the GAMS software can be excessively time-consuming. Consequently, a novel linear relaxation approach has been devised to enhance the efficiency of the solution process. This approach involves the transformation of the integer variables P_{ik}, R_{km}, and Q_{mj} into continuous variables, thereby yielding considerable reductions in computation time. Since the linearized objective function value is generally lower than the integer objective function value in minimization models ($\min LP \leq \min IP$), the solution obtained from the linear relaxation approach serves as a lower bound for the proposed model. However, there is a significant gap between the optimal solution obtained from the linear relaxation approach and the optimal solution obtained from the CPLEX solver in the GAMS software in the integer state of the variables P_{ik}, R_{km}, and Q_{mj}. As a result, a linear relaxation approach has been employed to effectively reduce the problem solution space. The procedure begins with solving the problem using the linear relaxation approach. Subsequently, the hubs identified through the linear relaxation approach are fixed within the proposed mixed-integer mathematical model. This approach plays a crucial role in determining specific nodes as hubs within the network in a shorter time frame. In essence, the linear relaxation approach significantly contributes to minimizing the solution space and expediting the problem-solving process. The subsequent step involves solving the mathematical model in the integer state of P_{ik}, R_{km}, and Q_{mj} variables by fixing certain nodes identified as hubs through the linear relaxation approach while leaving the remaining network nodes unconstrained. This is accomplished by executing the model again using the CPLEX solver in the GAMS software. The step-by-step execution of the proposed approach is illustrated in Fig. 5, outlining the process of fixing selected nodes obtained from the linear relaxation approach as hubs and leaving the other network nodes free.

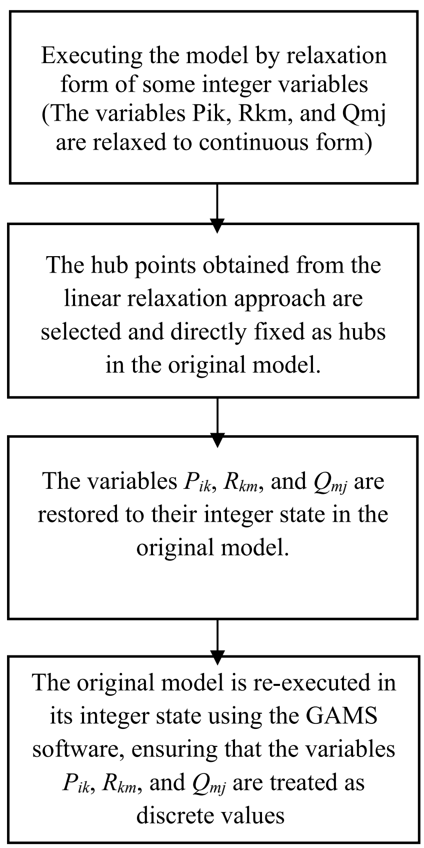

Fig. 5. The sequence of steps in performing the proposed linear relaxation algorithm.

4.2. Designing a Genetic Algorithm for Solving Large-Scale Problems

The Genetic Algorithm (GA) is a powerful search algorithm that is widely employed in various fields to find near-optimal solutions within large solution spaces. It draws inspiration from the principles of population genetics and was first introduced by Holland (1975). Since then, the GA has gained significant popularity and has been extensively applied to solve a diverse range of complex problems documented in literature. For detailed information and a comprehensive understanding of the GA, two highly recommended sources are the books authored by Mitchell (1998) and Haddow and Tufte (2010). These authoritative references provide in-depth explanations of the theory and practical applications of the GA. They serve as valuable resources for researchers and practitioners seeking to explore and leverage the potential of the GA in optimizing different types of problems.

In this section, a detailed description of the proposed algorithm is provided, starting with the generation of chromosomes based on the input data. The algorithm follows a structured process of genetic evolution, resulting in the generation of solutions for the problem. The fitness function is employed to evaluate the suitability of candidate solutions, and specific operators are used for selection and modification within the algorithm. The algorithm incorporates an exit condition that determines when the problem has been adequately addressed. Specifically, the algorithm terminates after a fixed number of iterations, which is set to 100 in this case. During the chromosome generation stage, a population is formed by selecting hub and non-hub nodes based on the distance matrix and node weights. The nodes are sorted in descending order of their weights, and nodes with high weights are chosen as hubs, while the remaining nodes are designated as non-hubs. To represent the problem, the algorithm utilizes two types of solutions. Separate operators for crossover and mutation are employed to manipulate these solutions during the evolutionary process.

The GA used in the proposed model can be defined as follows:

4.2.1. Initial Population Generation

The generation of the initial population involves creating a set of individuals that serve as the starting point for the GA. This process is crucial in establishing the diversity and exploration capabilities of the algorithm. To generate the initial population, the size of the population, denoted as npop, is determined based on the dimensions of the N node matrix (d, w). The population is generated randomly, considering the selection of solution matrices. In this context, two populations are created to match the dimensions of the network node matrix. The generation process entails randomly assigning nodes as hubs or non-hubs, forming a matrix that represents the initial population. The number of individuals in the initial population can vary depending on the specific problem and desired algorithm performance. For instance, in this case, a population size of 1000 has been considered. As the algorithm progresses through iterations, additional optimal solutions are generated using combination and mutation operators. These new solutions replace the existing population, ensuring that the algorithm explores different potential solutions at each iteration. The process involves preserving a set of 1000 optimal solutions in each iteration while discarding the remaining solutions.

By generating an initial population that encompasses diverse solutions, the GA can effectively commence the evolutionary process, facilitating the search for better solutions to the problem at hand.

4.2.1 Fitness Function

The fitness function is designed to optimize the computational efficiency of the algorithm by considering two flow path scenarios for the embedded problem. In one scenario, where the hubs are represented by binary values (0 and 1), the model operates based on the discrete conditions of data transfer between nodes. In the other scenario, where the nodes can take values between 0 and 1, the population can select values above 0.5 as hubs, while the remaining nodes are considered non-hubs.

To address the importance of the fitness function in evaluating solution quality, a sub-problem is formulated within the main problem. This involves considering two flow path configurations, each corresponding to the variable X(i, j, k, m). By representing the flow path as X(i, j, k, m) and its complement (1-X(i, j, k, m)), the goal is to achieve accurate results in minimal time while ensuring a suitable balance between discrete and continuous flow along the edges. This approach is facilitated by forming a population with a matrix that aligns with the network node structure.

The model comprises four components: the first component represents the cost of hub construction, the second component represents the cost of small aircraft flights from non-hub nodes to hubs, the third component represents the cost of large aircraft flights between hub nodes, and the fourth component represents the cost of small aircraft flights from hubs to non-hub nodes. Additionally, constraints related to flow paths are incorporated as constraints 4, 5, and 6 in the original problem, denoted as P, Q, and R, respectively. These constraints are added to the objective function to guide the optimization process effectively. In the proposed algorithm, two hubs, k1 and m1, are assigned to the population P1. This assignment is made considering the variable number of flights in the objective function, where the limit for the number of flights from non-hub node i to hub $k1$ is defined as constraint 4 in the model.

In the main problem, this constraint is calculated as follows and represented by the P_{ik} variable in the main objective function.

$$P_{ik} = Ceil\left(\frac{P_{ik1} + (w_{ij} \times X_{ijkm})}{Cap^{small}}\right) \tag{10}$$

Similarly, utilizing constraints 5 and 6, the variables R_{km} (flights from hub k1 to hub m1) and Q_{mj} (flights from hub m1 to non-hub j) are calculated according to relations 11 and 12, respectively.

$$R_{km} = Ceil\left(\frac{R_{k1m1} + (w_{ij} \times X_{ijkm})}{Cap^{large}}\right) \tag{11}$$

$$Q_{mj} = Ceil\left(\frac{P_{m1j} + (w_{ij} \times X_{ijkm})}{Cap^{small}}\right) \tag{12}$$

In this sub-problem, two additional hubs, K2 and m2, are introduced as a separate path in the population P2. Their purpose is to calculate the variable number of flights/transports from the non-hub node i to the hub node k2 in the objective function.

$$P_{ik} = Ceil\left(\frac{P_{ik2} + (w_{ij} \times (1 - X_{ijkm}))}{Cap^{small}}\right) \tag{13}$$

Consequently, to calculate the variables R_{km} (flights from hub k2 to hub m2) and Q_{mj} (flights/transports from hub m2 to non-hub j) in the objective function, the following procedure is followed:

$$R_{km} = Ceil\left(\frac{R_{k2m2} + (w_{ij} \times (1 - X_{ijkm}))}{Cap^{large}}\right) \tag{14}$$

$$Q_{mj} = Ceil\left(\frac{P_{m2j} + (w_{ij} \times (1 - X_{ijkm}))}{Cap^{small}}\right) \tag{15}$$

In this research, the fitness value of the population, which is represented as a matrix derived from the chromosomes, was examined as the negative value of the objective function of the problem. Constraint violations were penalized based on a relative infeasibility degree. If a solution falls outside the constraints, a relatively large penalty value is added to the objective function to ensure that the final solution is penalized for such violations. By incorporating this penalty mechanism, the optimization process aims to prioritize feasible solutions that satisfy the problem's constraints. The higher penalty value for constraint violations discourages the algorithm from converging towards infeasible solutions and encourages exploration of feasible regions in the solution space.

4.2.2 Combination and Crossover Operator

The crossover operator plays a crucial role in GA by facilitating the recombination of genetic material from two parent chromosomes to produce offspring chromosomes. This process aims to generate improved chromosomes and enhance the overall quality of the population. During the crossover operation, the genetic information of two selected parent chromosomes is combined to create new chromosomes for the current generation. This recombination process involves merging the genes from the parent chromosomes, resulting in the creation of novel chromosomes within the current population. The crossover process is iteratively performed across all generations of the GA. Typically, chromosomes with higher fitness are favored, and a greater number of offspring copies are produced from these chromosomes. Consequently, a mating pool is formed at the end of the reproduction process, housing all the newly generated offspring chromosomes. The parent chromosomes participating in the crossover operation are chosen based on their fitness, often employing selection mechanisms such as fitness-proportionate selection or tournament selection. The offspring chromosomes produced through crossover are considered the next generation's potential solutions. In GA, the crossover probability parameter, denoted as Pc, determines the percentage of the population subjected to the crossover operator. The remaining percentage, (1-Pc), retains the unchanged chromosomes in their original form within the current generation. In this specific GA employed in the study, parent chromosomes are selected based on their fitness performance. The crossover operator is executed with a predefined rate of 0.25, indicating that a quarter of the population undergoes crossover to generate new solutions. Given an initial population size of 1000, this results in the creation of 250 fresh offspring solutions through the crossover operation. In this study, to promote diversity and exploration within the population, three different crossover operators are employed: one-point crossover, two-point crossover, and uniform crossover. These operators offer distinct strategies for combining genetic information between parent chromosomes.

- In the one-point crossover operator, a random point is selected within the solution structure matrix of the parent chromosomes. The genes beyond this point are swapped between the parents, creating offspring chromosomes with a combination of genetic material from both parents.
- In the two-point crossover operator, two random points are chosen within the solution structure matrix of the parent chromosomes. The genes located between these points are exchanged between the parents, resulting in offspring chromosomes that possess a mix of genetic information from both parents.
- The uniform crossover operator, with a probability of 0.5, exchanges all related genes between the parent chromosomes. This operator offers a more comprehensive mixing of genetic material, leading to offspring chromosomes that inherit a diverse combination of genes from the parents.

According to Fig. 6, the developed algorithm for the proposed model uses two operators: the one-point crossover operator and the two-point crossover operator. In the one-point crossover operator, a random point is selected within the solution structure matrix from the two parents among the origin or destination nodes of the network. This random point is selected to introduce more diversity in the solutions, with a 50% probability of being chosen among the origin nodes and a 50% probability of being chosen among the destination nodes. All values after this point are replaced with the corresponding values from the other parent. In the two-point crossover operator, two random points are chosen within the solution structure matrix from the two parents among the origin or destination network nodes. The related genes between these points are exchanged in the parents. In the uniform crossover operator, with a probability of 0.5, all the related genes of each solution in the parents are exchanged.

$$\rightarrow \begin{bmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{bmatrix} \rightarrow \begin{bmatrix} X'_{11} & X'_{12} & X'_{13} & X'_{14} & X'_{15} & X'_{16} \\ X'_{21} & X'_{22} & X'_{23} & X'_{24} & X'_{25} & X'_{26} \\ X'_{31} & X'_{32} & X'_{33} & X'_{34} & X'_{35} & X'_{36} \\ X'_{41} & X'_{42} & X'_{43} & X'_{44} & X'_{45} & X'_{46} \\ X'_{51} & X'_{52} & X'_{53} & X'_{54} & X'_{55} & X'_{56} \\ X'_{61} & X'_{62} & X'_{63} & X'_{64} & X'_{65} & X'_{66} \end{bmatrix}$$

$$\downarrow$$

$$\begin{bmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X'_{41} & X'_{42} & X'_{43} & X'_{44} & X'_{45} & X'_{46} \\ X'_{51} & X'_{52} & X'_{53} & X'_{54} & X'_{55} & X'_{56} \\ X'_{61} & X'_{62} & X'_{63} & X'_{64} & X'_{65} & X'_{66} \end{bmatrix} \quad \begin{bmatrix} X'_{11} & X'_{12} & X'_{13} & X'_{14} & X'_{15} & X'_{16} \\ X'_{21} & X'_{22} & X'_{23} & X'_{24} & X'_{25} & X'_{26} \\ X'_{31} & X'_{32} & X'_{33} & X'_{34} & X'_{35} & X'_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{bmatrix}$$

Fig. 6. One-point combination operator

4.2.3. Mutation Operator

In this mutation operator, new information is randomly incorporated into the search process of the GA. This crucial feature assists the GA in escaping local optima and increasing diversity in the population of chromosomes or candidate solutions. It results in significant alterations in the offspring chromosomes, causing them to have entirely different genes compared to their parent chromosomes. Essentially, when genes are copied from the current chromosome to the new chromosome, there is a probability for each gene to undergo a mutation. This probability, commonly referred to as the mutation probability (Pm), is usually a very small value. In this study, the mutation operator is defined with a predetermined rate of 0.15. For the purposes of this research, a random mutation is employed, executed by randomly selecting parents, which leads to the swapping of node positions. This, in turn, induces a probabilistic change in the location of the hub nodes in the network. Throughout this process, the gene values for these chromosomes are altered with a probability of 0.2, as illustrated in Fig. 7.

$$\begin{array}{l} \\ \rightarrow \\ \\ \rightarrow \\ \\ \\ \end{array} \begin{bmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X'_{31} & X'_{32} & X'_{33} & X'_{34} & X'_{35} & X'_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X'_{51} & X'_{52} & X'_{53} & X'_{54} & X'_{55} & X'_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{bmatrix}$$

Fig. 7. Random Mutation Operator

4.2.4. Selection Operator

The crossover and mutation operators are responsible for propagating one generation to the next in a GA. In order to accomplish this, the best solutions of npop from the previous generation and the newly generated offspring are retained based on their fitness functions using a roulette wheel selection mechanism. This ensures that the next generation takes shape by incorporating superior individuals from both the parent generation and the offspring.

4.2.5. The Termination Condition

The termination condition of the designed GA mandates that after a predefined number of iterations, the algorithm reaches its conclusion. In this case, the algorithm is set to terminate after 100 iterations, ensuring that an adequate number of generations are processed to potentially yield desirable solutions. The flowchart of the GA is shown in Fig. 8.

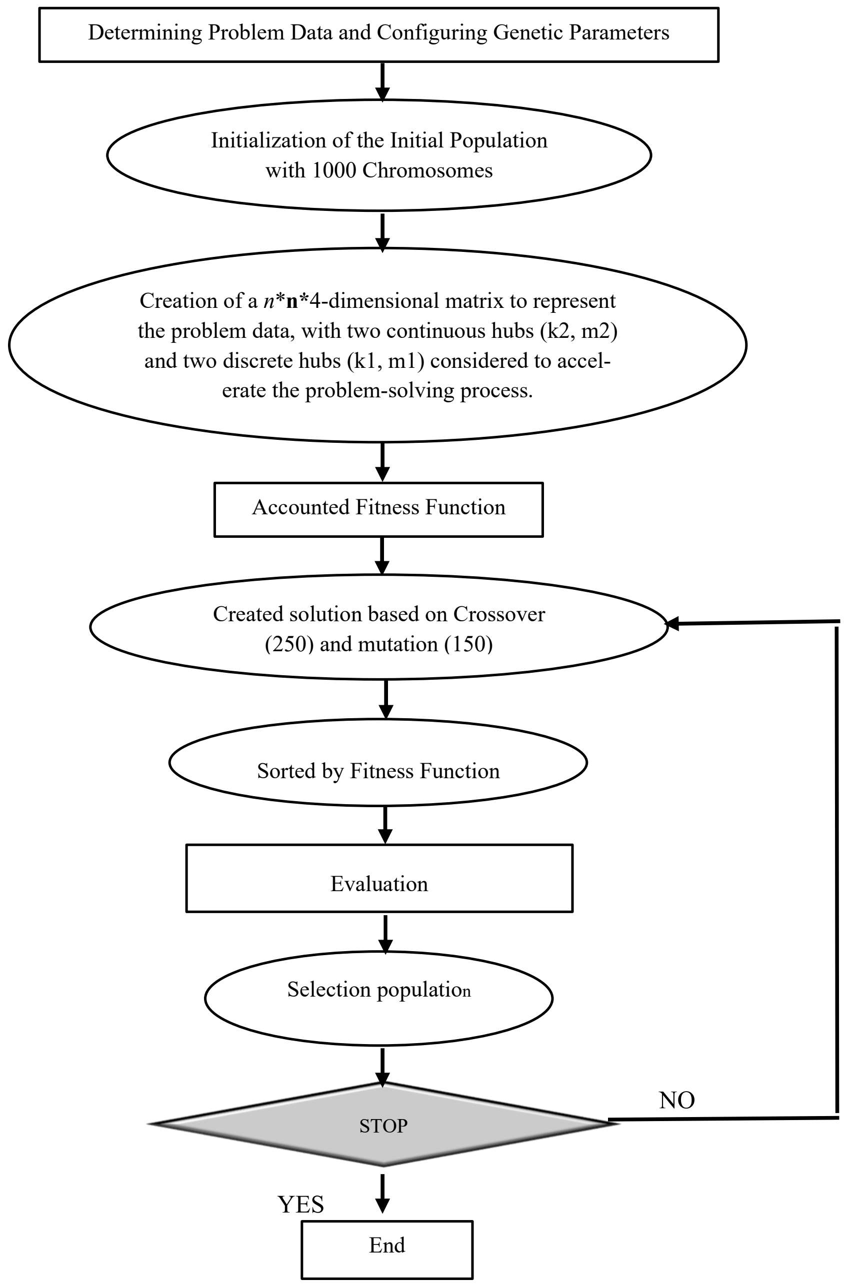

Fig. 8. Flowchart of The Proposed Genetic Algorithm

The pseudo-code of the proposed GA is presented in figure 9 as below:

```
Initialization:
    Set GA Parameters: Crossover Probability (p_cr); Mutation Probability (p_mu); Population Size (P); and Maximum Iteration (Iter)
        Iter ← 0
        Generate Initial Population (Randomly): P (Iter)
        Evaluate the Initial Population to Calculate Costs of All Nodes (For The
        Hub Selection Problem): P (Iter)
    Do while (till Iter = Maximum Iteration)
        Iter ←Iter+1
        Choose Two Parents Randomly from P (Iter-1)
        Generate A Random Number (R1) Between 0 And 1
        If R1 < p_cr Then
            Perform Crossover (Recombine Method)
            Perform Mutation (Randomly Mehthod)
            Evaluate Offsprings: O (Iter)
            If Offspring's Costs Objective Is Improved Upon Mutation Then
                Add the Mutated Offspring to The New Population
            Else
                Add the Crossovered Offspring to The New Population
            End If
        Else if
            Select One Of The Two Parents Randomly
            Generate A Random Number (R2)
            If R2 < p_mu Then
                Perform Mutation
                If Chromosome's or Parents Costs Objective Is Improved Upon Mutation Then
                Add the Mutated Offspring to The New Population
                End If
            Else
                Add the Parent to The Population
            End If
        Else
            Add the Selected Parent to The New Population
        End If
    Rank the Parents in Population P (Iter).
    Rank the Offspring Population O (Iter).
    Insert the Superior Members of P (Iter) And O (Iter) Into P (Iter+1).
    Evaluate P (Iter+1)
    Loop Till
    Iter ← Maximum Iter
    End while
```

Fig. 9. pseudo-code of Genetic Algorithm

4.3. Taguchi Method for Tuning Genetic Algorithm Parameters

This section outlines the methodology employed for effectively tuning the parameters of a genetic algorithm using the Taguchi Method. The goal is to achieve optimal parameter values that enhance the algorithm's performance and efficiency. The approach involves three key steps: orthogonal array (OA) design, scenario generation and repetition, and evaluation metrics.

4.3.1. Methodology

- Orthogonal Array (OA) Design: We construct a matrix of factors representing different levels of the algorithm parameters. This matrix is carefully designed according to the principles of the Taguchi Method. By utilizing orthogonal arrays, we ensure a balanced and efficient exploration of parameter combinations.
- Scenario Generation and Repetition: In order to comprehensively evaluate the impact of different parameter settings, nine distinct scenarios are defined. Each scenario undergoes five repetitions, allowing us to obtain reliable

and statistically significant results. This step is crucial for capturing the variability and robustness of the algorithm's performance across different parameter values.

- Evaluation Metric: The algorithm's performance under each parameter setting is measured using the signal-to-noise ratio (S/N). The S/N ratio, as described in equation 16, quantifies the average desired response value relative to the standard deviation of the undesired value. By maximizing the calculated S/N, we aim to identify the parameter configuration that achieves the optimal trade-off between the desired response and variability.

$$S = -10 \log_{10} \left(\frac{Z_{ij}^2}{n} \right) \quad \text{for i, j} \tag{16}$$

The values considered for the parameters in GA are shown in Table 1:

Table 1
The value of the parameters

	1	2	3
Crossover Percentage	0.25	0.5	0.95
Mutation Percentage	0.15	0.05	0.01
Population Size	10	50	100
Number of Iterations	50	100	200
Mutation Rate	0.2	0.5	0.8
Selection Pressure Rate	0.001	0.0001	0.00001

Table 2 represents the nine different scenarios considered to examine each parameter level:

Table 2
The nine different scenarios to examine each parameter level

Scenarios					
1	1	1	1	1	1
1	2	3	2	2	2
1	3	2	3	3	3
2	1	3	3	3	3
2	2	2	1	1	1
2	3	1	2	2	2
3	1	3	2	2	
3	2	1	3	3	
3	3	3	1	1	

The pseudo-code for the proposed Taguchi method used for tuning the parameters of the proposed genetic algorithm is as Fig. 10:

Producer: Taguchi design for GA parameters
Input: P cross, P Mutation, Pop Size, Max Iteration, Mu
Output: optimum level of parameter
Begin
    Select L_9 (3^4) as the suitable OA
    Apply GA on each scheme of L_9 (3^4)
    Obtain fitness for each scheme
        For i 1 to 4 do
            For j 1 to 3 do
                Calculate S/N ratio, make span, PRD
            End
        End
Determine optimum level of parameters
End

Fig. 10. pseudo-code of Taguchi Method for Genetic Algorithm

5. Results

5.1. Dataset Definition

The selected dataset for evaluating the performance of the modeling model consists of two datasets: the dataset of passenger aircraft in the United States and the cargo dataset in Turkey. The dataset of passenger aircraft in the United States is based on information regarding the distance and passenger flow of airlines in 25 major cities in the United States in the year 1970. This dataset was initially used in O'Kelly's studies. The TR dataset, on the other hand, contains data on the 81-node Turkish network. It provides information about various aspects of the Turkish transportation network. Both datasets are utilized in this study to assess and analyze the performance of the modeling model. In this section, the value of the fixed cost parameter is obtained from previous research for both datasets, while other necessary data are derived from an analysis of Tables 1 to 4 for the CAB dataset. Five different values for the fixed hub construction cost have been considered.

$$f_k = \{50, 100, 150, 200, 250\}$$

The cost of utilizing large aircraft between hubs for each flight (FCR), as well as the cost of utilizing small aircraft between non-hub origins and hubs (FCP) and hubs and non-hub destinations for each flight (FCQ), have been factored in based on the values presented in the following table. These costs are taken into consideration relative to the cost ratio of establishing hubs, given that the cost of establishing hubs is several times greater than the fixed transportation cost.

$$\{FCP, FCR, FCQ\} = \{10, 20, 10\}$$

In the proposed model, the aircraft capacity for passenger transportation from non-hub nodes to hubs and from hubs to non-hub nodes is considered Cap^{small}, while the aircraft capacity for passenger transportation between hubs is considered Cap^{large}. If the aircraft capacities are treated as variables, the mathematical model would become nonlinear, as the right-hand side of constraints (4), (5), and (6) would involve the product of the variable quantities of transportation vehicles and the variable capacity of the vehicles. Therefore, to prevent the problem from becoming nonlinear, the optimal capacities of the transportation vehicles are parametrically considered by performing calculations based on the table below.

Table 3
Determining the Capacity of Transportation Vehicles between Hubs and between Non-Hub to Hubs (n=10)

Cap^{small}, Cap^{large}	f_k=50, FCP=10, FCQ=10, FCR=20 Hubs	Objective function value	f_k=200, FCP=10, FCQ=10, FCR=20 Hubs	Objective function value
500, 1000	1,2,3,4,5,6,7,8,9,10	1484.668	4,5,6,7,9,10	2500.970
1500, 3000	3,4,7,9	563.113	4,9	984.257
2500, 5000	4,6,7	387.463	7,9	710.518
3000, 6000	7,9	340.059	7,9	640.059

Table 4
Determining the Capacity of Transportation Vehicles between Hubs and between Non-Hub to Hubs (n=15)

Cap^{small}, Cap^{large}	f_k=50, FCP=10, FCQ=10, FCR=20 Hubs	Objective function value	f_k=200, FCP=10, FCQ=10, FCR=20 Hubs	Objective function value
1500, 3000	1, 3, 4, 7, 8, 9, 12, 14, 15	1522.423	5, 7, 9, 11	2443.030
2500, 5000	3, 4, 7, 8, 9, 12, 14	986.580	4, 6, 7	1617.960
3000, 6000	4, 6, 7, 12, 14	846.018	5, 11	1380.210
4000, 8000	4, 6, 7, 12, 14	689.079	4, 7	1103.850

Table 5
Determining the Capacity of Transportation Vehicles between Hubs and between Non-Hub to Hubs (n=20)

Cap^{small}, Cap^{large}	f_k=50, FCP=10, FCQ=10, FCR=20 Hubs	Objective function value	f_k=200, FCP=10, FCQ=10, FCR=20 Hubs	Objective function value
3000, 6000	1, 3, 4, 6, 7, 8, 12, 14, 17, 18	1752.558	4, 6, 7, 12, 17, 18	2833.182
4000, 8000	4, 6, 7, 12, 14,17, 18	1341.132	4, 6, 12, 17	2203.680
5000, 10000	4, 7, 12, 14, 17, 20	1107.772	6, 12, 13, 17	1826.336
6000, 12000	4, 7, 12, 14, 17, 20	991.953	6, 7, 17	1556.752

Table 6
Determining the Capacity of Transportation Vehicles between Hubs and between Non-Hub to Hubs (n=25)

Cap^{small}, Cap^{large}	f_k=50, FCP=10, FCQ=10, FCR=20 Hubs	Objective function value	f_k=200, FCP=10, FCQ=10, FCR=20 Hubs	Objective function value
4000, 8000	1, 4, 7, 8, 9, 12, 14, 17, 18, 22, 25	2005.304	6, 12, 17, 21, 25	3113. 096
5000, 10000	4, 7, 9 12, 14, 17, 22, 25	1639.052	6, 12, 18, 21, 24	2547.590
6000, 12000	4, 6, 7, 12, 14, 17, 22, 25	1412.582	12, 17, 20, 21	2173.419
7500, 15000	4, 7, 12, 14, 17, 25	1116.512	12, 17, 20, 21	1840.067

The analysis of vehicle capacity for the CAB dataset is based on the data extracted from Table 3. The optimal configuration for the CAB dataset with 10 nodes entails a meticulous evaluation of the fixed costs associated with hub establishment. With fixed costs set at 50 and 200, the proposed model favors a small aircraft capacity of 1500 and a large aircraft capacity of 3000. This capacity allocation strikes a balance by generating a more suitable number of hubs compared to alternative capacity options for efficient transportation. Drawing from the insights provided in Table 4, the CAB dataset with 15 nodes requires a comprehensive analysis of fixed costs for hub creation. Setting the fixed costs at 50 and 200, the model recommends a small aircraft capacity of 2500 and a large aircraft capacity of 5000. This configuration optimizes hub distribution, resulting in a more desirable number of hubs when compared to other capacity options for efficient transportation. Furthermore, referencing Table 5, the CAB dataset with 20 nodes demands a thorough examination of fixed costs for hub establishment. With fixed costs set at 50 and 200, the proposed model advocates for a small aircraft capacity of 5000, while the capacity of the large aircraft remains unspecified. This strategic allocation of resources leads to the creation of a more favorable number of hubs, surpassing the performance of alternative capacity options. Table 6, which pertains to calculating the aircraft capacity for a dataset of 25 nodes in the CAB data, reveals insightful results. The obtained data demonstrates that utilizing fixed costs of 50 and 200 for hub creation yields a capacity of 75000 for the smaller aircraft and a capacity of 15000 for the larger aircraft. These values correspond to a reasonable number of hubs within the network. The calculations presented in the tables illustrate a notable trend: as the number of network nodes increases, the vehicle capacity required to establish a suitable number of hubs also rises. Additionally, it is evident that increasing the vehicle capacity (in this case, aircraft) reduces the number of hubs created within the network. Moreover, the location of hubs may vary under specific circumstances. This highlights the influential role of vehicle capacity in determining both the number and placement of hubs. In conclusion, these analyses emphasize the importance of considering vehicle capacity when determining the optimal number and locations of hubs within the network.

5.2 Results of Computational Calculations

Table 7 showcases the results of solving a model with 10 nodes from the CAB dataset, utilizing the CPLEX solver integrated into the GAMS software. It is evident that achieving the optimal solution for this specific node size required a considerable amount of time. Extrapolating this to larger node sizes implies that resolving the problem would demand a substantial time investment. However, an intriguing finding emerges from Table 7. By incorporating a 1% gap in the CPLEX solver, the time required to solve the problem is significantly reduced. Furthermore, the solutions obtained from the software with the 1% gap closely mirror those obtained without considering the gap. Remarkably, the disparity between the two solution sets is practically negligible, with the resulting hub locations being identical in both cases.

Table 7
Results of CAB dataset in (n=10, *CAP Small*= 1500, *CAP Large*=3000, *FCP*, *FCQ* =10, *FCR*=20)

	Optimal solution by GAMS			Best Known Solution by GAMS with 1% GAP form Optimality			Gap (%)
f_k	**Hubs**	**OPT**	**Time(S)**	**Hubs**	**OPT**	**Time(S)**	
50	3,4,7,9	563.11	5418	3,4,7,9	563.71	31	0.1
100	4,7,9	726.90	429	4,7,9	727.42	3	0.07
150	4,6,7	876.53	1437	4,6,7	877.51	4	0.11
200	4,9	984.25	4288	4,9	984.30	4	0.005
250	4,9	1084.25	4142	4,9	1084.70	2	0.04

The parameters obtained using the Taguchi method for the calculations in Tables are as follows: beta = 0.001, Mu = 0.2, P Crossover = 0.95, P mutation = 0.01, Pop Size = 100, Iteration = 50. These parameter values for 10, 15, and 20 Nodes with different fixed cost and flight capacity are illustrated in Figure 11 to 16.

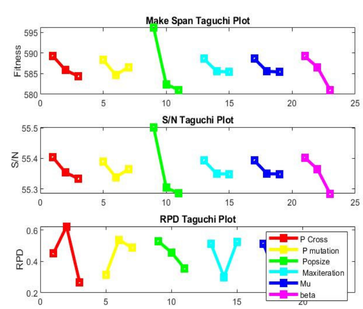

Fig. 11. Parameter Optimization for 10 Nodes with a Fixed Cost of 50 and Flight Capacity of 1500 and 3000.

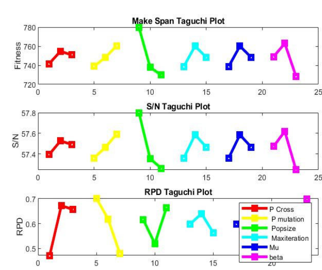

Fig. 12. Parameter Optimization for 10 Nodes with a Fixed Cost of 100 and Flight Capacity of 1500 and 3000.

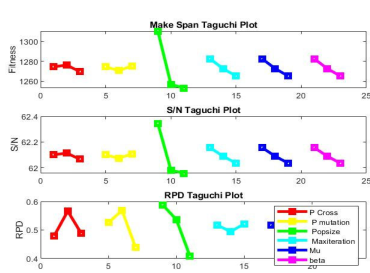

Fig. 13. Parameter Optimization for 10 Nodes with a Fixed Cost of 100 and Flight Capacity of 2500 and 5000.

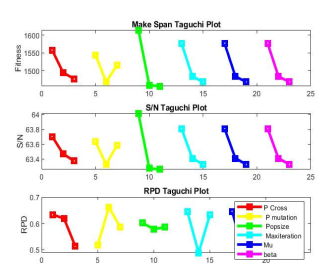

Fig. 14. Parameter Optimization for 10 Nodes with a Fixed Cost of 150 and Flight Capacity of 2500 and 5000.

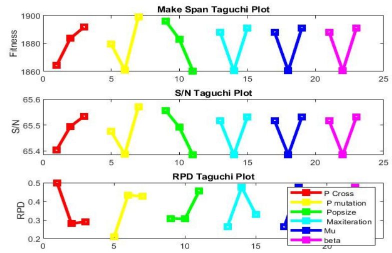

Fig. 15. Parameter Optimization for 20 Nodes with a Fixed Cost of 200 and Flight Capacity of 5000 and 10000.

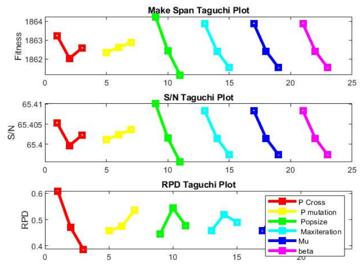

Fig. 16. Parameter Optimization for 20 Nodes with a Fixed Cost of 250 and Flight Capacity of 5000 and 10000.

Table 8
Results of CAB dataset in (n=10, *CAP Small*= 1500, *CAP Large*=3000, *FCP*, *FCQ* =10, *FCR*=20)

f_k	Best Known Solution by GAMS			Obtained Solution by LP Relaxation Based Method			Gap (%) Between LP Relaxation and GAMS	Obtained Solution by GA parameters Tuned by Taguchi			Gap (%) GA Vs. GAMS
	Hubs	OPT	Time(S)	Hubs	OPT	Time(S)		Hubs	OPT	Time(S)	
50	3,4,7,9	563.11	31	3,4,6,7	566.14	14	0.5	4,6,7,8	579.798	30.364	2.8
100	4,7,9	726.90	3	4,7,9	726.90	22	0	4,7,8	727.90	30.449	0
150	4,6,7	876.53	4	4,6,7	876.95	5	0.04	4,6,7	876.53	30.082	0
200	4,9	984.25	4	4,9	984.25	31	0	4,9	984.25	29.463	0
250	4,9	1084.25	2	4,9	1084.25	8	0	4,9	1174.45	27.452	1. 2

Table 9
Results of CAB dataset in (n=15, *CAP Small*= 2500, *CAP Large*=5000, *FCP*, *FCQ* =10, *FCR*=20)

f_k	Best Known Solution by GAMS			Obtained Solution by LP Relaxation Based Method			Gap (%) Between LP Relaxation and GAMS	Obtained Solution by GA parameters Tuned by Taguchi			Gap (%) GA Vs. GAMS
	Hubs	OPT	Time(S)	Hubs	OPT	Time(S)		Hubs	OPT	Time(S)	
50	3,4,7,9,12,14	986.58	195	4,6,7,8,12,14	986.89	60	0.03	1,4,6,7,8,12	996.80	56.73	1
100	4,7,9,12,14	1240.45	36	4,7,9,12,14	1240.16	51	0.03	1,4,6,7,12	1254.25	55.91	1.1
150	4,7,9,12	1455.01	201	4,7,9,12	1452.54	55	0.17	1,4,7,12	1482.48	54.12	1.8
200	4,6,7	1617.96	143	4,12,13	1620.40	70	0.15	1,4,7,12	1636.04	53.90	1.1
250	9,12,13	1766.76	144	4,7,9	1761.47	125	0.3	1,4,7,12	1835.34	54.01	3.7

Table 10
Results of CAB dataset in (n=20, *CAP Small*= 5000, *CAP Large*=10000, *FCP*, *FCQ* =10, *FCR*=20)

f_k	Best Known Solution by GAMS			Obtained Solution by LP Relaxation Based Method			Gap (%) Between LP Relaxation and GAMS	Obtained Solution by GA parameters Tuned by Taguchi			Gap (%) Vs. GA and GAMS
	Hubs	OPT	Time(S)	Hubs	OPT	Time(S)		Hubs	OPT	Time(S)	
50	4,7,12,14,17,20	1107.77	528	4,6,7,12,17,18	1107.86	488	0.11	4,6,7,12,14,17	1112.10	99.656	0.3
100	4,7,12,17,20	1406.96	664	4,7,12,17,20	1406.56	657	0.02	1,4,7,12,17	1446.14	96.104	2.7
150	9,12,13,17	1632.55	661	4,12,17,20	1630.18	411	0.14	1,4,7,12,17	1695.44	96.707	3.7
200	6,12,13,17	1827.75	2442	4,12,17,20	1829.36	965	0.08	1,4,17,19	1868.64	96.424	2.1
250	6,7,18	1991.77	2185	4,12,17,20	2029.28	1029	1.8	1,4,17,19	2069.19	97.484	3.7

Table 11
Results of CAB dataset in (n=25, *CAP Small*= 7500, *CAP Large*=15000, *FCP*, *FCQ* =10, *FCR*=20)

f_k	Best Known Solution by GAMS			Obtained Solution by LP Relaxation Based Method			Gap (%) Between LP Relaxation and GAMS	Obtained Solution by GA parameters Tuned by Taguchi			Gap (%) Vs. GA and GAMS
	Hubs	OPT	Time(S)	Hubs	OPT	Time(S)		Hubs	OPT	Time(S)	
50	4,7,12,14,17,25	1163.51	4461	4,7,12,14,17,25	1162.91	4354	0.05	4,6,7,12,14,17	1217.44	150.80	4.4
100	12,17,20,21	1439.87	9960	1,4,12,17	1447.17	5312	0.5	1,4,12,17	1444.66	151.70	0.3
150	6,12,17,21	1641.01	8728	4,12,17,25	1647.26	7014	0.37	1,4,12,17	1642.12	143.42	0.07
200	12,17,20,21	1840.06	20320	1,4,12,17	1847.71	13326	0.4	1,4,12,17	1847.72	144.07	0.4
250	2,5,12	1999.70	18029	5,12,20	2019.01	12829	0.95	2,6,12	2013.83	147.83	0.7

5.3 Analysis of the results

As the number of variables in a problem increases, the solution space expands correspondingly. Consequently, solving the problem becomes more time-consuming, especially for larger sizes. The results of solving the problem for sizes 15, 20, and 25 using the GAMS software are presented in Tables 8, 9, 10, and 11, respectively. To address the time-consuming challenge, the proposed model employed two approaches. Firstly, an exact method called the LP relaxation-based method was utilized. Additionally, meta-heuristic algorithms, specifically the GA, were implemented. These algorithms were employed to explore various solution paths and optimize the results. The solutions obtained from the LP relaxation-based method and the meta-heuristic algorithm were compared with results obtained from the CPLEX solver in GAMS to evaluate their effectiveness and determine the most promising solution. Tables 5 to 9 demonstrate that the percentage difference between the LP relaxation-based method's results and those obtained from the GAMS software is consistently below one percent, approaching zero in most cases. In only one case, the GAP is 1.8 percent. Some of the results obtained from the LP relaxation-based method show only slight enhancements compared to the results obtained from GAMS. It is important to note that if the 1% difference criterion was not applied in the software, due to the substantial time required to run the model, all results obtained by GAMS would either be equal to or better than the LP relaxation-based method's results. However, the tables clearly show that the proposed method significantly reduces the computational time compared to the software solution. In some cases, the proposed method achieves a 50% reduction in computational time for networks with 20 and 25 nodes, which is a significant advantage of the proposed approach. Although the LP-based relaxation method proves effective for medium-sized problems, it may not be well-suited for larger-scale problems. As the problem size increases, the method becomes increasingly time-consuming and may struggle to solve excessively large problems. Consequently, a GA has been developed specifically for addressing the challenges posed by large-sized instances of the proposed model. The results presented in Tables 5–9 demonstrate that the GA is capable of finding solutions within a significantly reduced timeframe and exhibits minimal discrepancy when compared to the results obtained from GAMS with a below-4.5% gap with the GAMS solution. Therefore, the developed GA is employed to handle the TR datasets, which consist of 81 nodes. Tables 12 and 13 showcase the results obtained for the TR datasets using the GA.

Table 12
Results of Genetic Algorithm for vehicle capacity of 6000 and 12000 on TR dataset (n=81)

f_k	Hubs	OPT	Times(S)
50	1,6,7,21,26,25,20,34,45,44,53,54,60,68,80	6851.378	752.248
100	1,3,6,21,27,34,41,55,45,69,58,81	7582.671	1380.902
150	6,15,23,34,60,28,54,45,80	8298.722	1367.975
200	6,25,23,34,52,64,54,80	8726.745	1342.594
250	6,23,34,64,60,80	9143.963	1323.612

Table 13
Results of Genetic Algorithm for vehicle capacity of 10000 and 20000 on TR dataset (n=81)

f_k	Hubs	OPT	Times(S)
50	1,15,6,21,16,25,34,46,35,38,42,52,55,66,81	4447.08	1391.133
100	3,6,23,16,42,35,34,52,66,80	5090.439	1347.265
150	6,16,23,52,34,35,42,80	5516.337	1328.071
200	6,23,34,43,60,45,80	5864.336	1323.592
250	6,12,41,60,64,80	6124.553	1276.448

The results obtained from the tables reveal that as the cost of establishing a hub increases from 50 to 250, the number of hubs gradually decreases. This observation aligns with previous research on hub location issues and validates the model. With higher fixed costs for hub establishment, the model aims to minimize the total cost of the network by reducing the number of hubs. This reduction encompasses both the cost of hub creation and the cost of network transportation. Fig. 11 illustrates the hub locations and the allocation of other nodes to those hubs in a sample network comprising 25 nodes. The cost associated with creating a hub is set at 200 in this scenario.

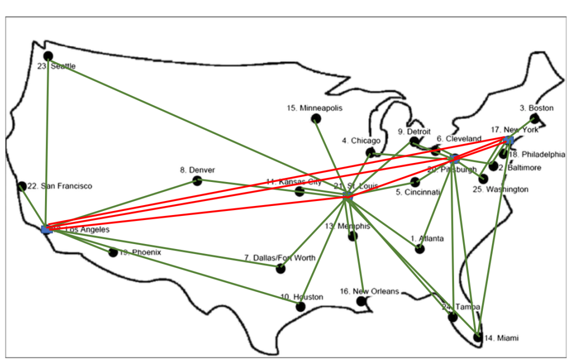

Fig. 17. Hub locations and non-hub nodes allocation in CAB (n=25) with a hub creation cost of 200

The proposed model involves multiple allocations of modular hub locations. In this mathematical model, the variable $\mathbf{X_{ijkm}}$ represents the flow route from a source node **i** to a destination node **j** through hubs **k** and **m**. After solving the problem, this variable takes continuous values between zero and one. A value of zero for $\mathbf{X_{ijkm}}$ indicates that the source node **i** is not connected to hub **k**. On the other hand, a value of one represents 100% of the flow from source node **i** to hub **k,m**. If the value is between zero and one, it means that a portion of the flow sent from source node **i** to destination node **j** goes through hubs **k and m,** and another portion goes through other hubs. To gain a better understanding of the concept, consider Figure 17 as an example. Suppose the objective is to route flow from node 1, which represents the city of Atlanta, to node 2, representing the city of Baltimore, using node 20 (the city of Pittsburgh) as a hub. If the value of $\mathbf{X_{1,2,20,20}}$ is 1, it means that all the flow (100% of the flow) from node 1 to node 2 is routed through hub 20. However, node 1 is assigned to two hubs, numbered 20 and 21. When sending flow from node 1 to node 6, a portion of the flow goes through hub 20 and another portion goes through hub 21. As calculated, the values for $\mathbf{X_{1,6,20}}$ and $\mathbf{X_{1,6,21}}$ are 0.821 and 0.179, respectively. This implies that 82.1% of the flow from node 1 to node 6 is routed through hub 20, while 17.9% of the flow is routed through hub 21.

Tables 14 to 16 provide calculations related to the number of flights in the network as follows:

Table 14
Number of flights from non-hub nodes to hubs (value of P_{ik}) in CAB with n=20 and f_k=200

	Hubs			
Nodes	12	17	20	21
1			18	23
2		16	8	
3		87		
4			43	100
5			21	2
6			43	
7	7			37
8	11			24
9			55	6
10	5			30
`11				29
13				17
14		22	52	5
15				36
16				27
18		51		
19	22			
22	73			
23	25			3
24			18	9
25		62	20	

Table 15
Number of flights between hubs (value of R_{km}) in CAB with n=20 and f_k=200

Hubs	**12**	**17**	**20**	**21**
12		8	9	48
17	6		55	8
20	8	44		57
21	48	8	6	

Table 16. Number of flights from to hubs non-hub (value of Q_{mj}) in CAB with n=20 and f_k=200

	Non-hubs												
Hubs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
12							6	11		5			
17		17	87										
20	22	7		43	21	43			56		1		
21	19			100	2		38	24	5	30	28	5	17

	Non-hubs												
Hubs	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**14**
12						22				73	24		
17	2				51						4	6	62
20	73											21	20
21	4	36	27	42									

Tables 14 to 16 display the number of flights between nodes in the network. Despite the symmetrical nature of the w_{ij} network flow data matrix, as evident from Tables 14 to 16, the number of flights between nodes is not symmetrical. Instead, it is evenly distributed based on the distance between the nodes within the flight network. As mentioned before, the allocation variable X_{ijk} is treated as a continuous variable, indicating that a portion of the flow between an origin and destination occurs through hubs *k* and *m,* while another portion passes through different hubs. This approach ensures a balanced distribution of traffic among the network hubs to calculate the minimum value of the objective.

6. Conclusion

This paper introduces a novel model for tackling the multiple-allocation modular hub location problem, taking into consideration the capacity of transportation vehicles. A key characteristic of this model is its effectiveness in meeting demands by strategically utilizing selected hubs within the network. To evaluate the performance of the computational model, the CAB and TR datasets were utilized, facilitating a comprehensive analysis of the problem. The proposed model accurately calculates the number of trips within the network by considering the capacity of the vehicles transported between routes. Furthermore, it clearly demonstrates the direct relationship between vehicle capacity and the number of hubs in the network. This is since vehicle capacity directly influences the overall number of network trips.

Recognizing the NP-hard nature of HLPs, this study proposes a fresh approach to handling the complexity of the solution space. This involved the development of a new method that leverages linear relaxation of integer variables, resulting in faster computation times compared to the GAMS software, particularly for medium-sized problems. By utilizing the initial solution, the linear relaxation method aims to refine and enhance the results in a more efficient manner. Since the LP relaxation-based method encounters challenges in handling large-sized instances of the proposed models, an additional solution approach was introduced to further enhance the analysis. In this study, a genetic algorithm was developed and integrated into the analysis, and the parameters in GA were tuned by the Taguchi method. The effectiveness of the GA tuned with the Taguchi method was demonstrated through its ability to produce satisfactory results within a short timeframe, even for large-scale problems. Notably, the GA tuned with the Taguchi method achieved a solution gap of less than 4%, indicating its efficacy in finding close-to-optimal solutions for the given problem instances. The results analysis clearly demonstrates that both proposed algorithms provide high-quality solutions while significantly reducing the time required compared to the results obtained from the CPLEX solver in GAMS software.

References

Abdinnour-Helm, S., & Venkataramanan, M. A. (1998). Solution approaches to hub location problems. *Annals of Operations research*, *78*(0), 31-50.

Alumur, S. A., Campbell, J. F., Contreras, I., Kara, B. Y., Marianov, V., & O'Kelly, M. E. (2021). Perspectives on modeling hub location problems. *European Journal of Operational Research*, *291*(1), 1-17.

Alumur, S. A., Nickel, S., Saldanha-da-Gama, F., & Seçerdin, Y. (2016). Multi-period hub network design problems with modular capacities. *Annals of Operations Research*, *246*, 289-312.

Alumur, S., & Kara, B. Y. (2008). Network hub location problems: The state of the art. *European journal of operational research*, *190*(1), 1-21.

Athamneh, R. A., Tanash, M., Hani, D. B., Rawshdeh, M., Alawin, A., & Albataineh, Z. (2023). Variable Neighborhood Search Algorithm for the Single Assignment Incomplete Hub Location Problem with Modular Capacities and Direct Connections. *Operations Research Perspectives*, *11*, 100286.

Bashiri, M., Golkarian, A., & Nikzad, E. (2018a). Design of a manufacturing hub with modular automation units. *Procedia Manufacturing*, *17*, 911-918.

Bashiri, M., Rezanezhad, M., Tavakkoli-Moghaddam, R., & Hasanzadeh, H. (2018b). Mathematical modeling for a p-mobile hub location problem in a dynamic environment by a genetic algorithm. *Applied Mathematical Modelling*, *54*, 151-169.

Basirati, M., Akbari Jokar, M. R., & Hassannayebi, E. (2020). Bi-objective optimization approaches to many-to-many hub location routing with distance balancing and hard time window. *Neural Computing and Applications*, *32*(17), 13267-13288.

Bhattacharjee, A. K., & Mukhopadhyay, A. (2023). An improved genetic algorithm with local refinement for solving hierarchical single-allocation hub median facility location problem. *Soft Computing*, *27*(3), 1493-1509.

Bryan, D. L., & O'kelly, M. E. (1999). Hub-and-spoke networks in air transportation: an analytical review. *Journal of regional science*, *39*(2), 275-295.

Campbell, J. F. (1994). Integer programming formulations of discrete hub location problems. *European journal of operational research*, *72*(2), 387-405.

Campbell, J. F., & O'Kelly, M. E. (2012). Twenty-five years of hub location research. *Transportation Science*, *46*(2), 153-169.

Carello, G., Della Croce, F., Ghirardi, M., & Tadei, R. (2004). Solving the hub location problem in telecommunication network design: A local search approach. *Networks: An International Journal*, *44*(2), 94-105.

Contreras, I., & O'Kelly, M. (2019). Hub location problems. *Location science*, 327-363.

Corberán, Á., Peiró, J., Campos, V., Glover, F., & Martí, R. (2016). Strategic oscillation for the capacitated hub location problem with modular links. *Journal of Heuristics*, *22*, 221-244.

Correia, I., Nickel, S., & Saldanha-da-Gama, F. (2018). A stochastic multi-period capacitated multiple allocation hub location problem: Formulation and inequalities. *Omega*, *74*, 122-134.
Damgacioglu, H., Dinler, D., Ozdemirel, N. E., & Iyigun, C. (2015). A genetic algorithm for the uncapacitated single allocation planar hub location problem. *Computers & Operations Research*, *62*, 224-236.
De Freitas, C. C., Aloise, D. J., Da Costa Fontes, F. F., Santos, A. C., & Da Silva Menezes, M. (2023). A biased random-key genetic algorithm for the two-level hub location routing problem with directed tours. *OR Spectrum*, *45*(3), 903-924.
Ernst, A. T., & Krishnamoorthy, M. (1996). Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location science*, *4*(3), 139-154.
Ernst, A. T., & Krishnamoorthy, M. (1999). Solution algorithms for the capacitated single allocation hub location problem. *Annals of operations Research*, *86*(0), 141-159.
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*(4), 1096-1109.
Fard, M. K., & Alfandari, L. (2019). Trade-offs between the stepwise cost function and its linear approximation for the modular hub location problem. *Computers & Operations Research*, 104, 358-374.
Haddow, B. P., & Tufte, G. (2010). Goldberg DE genetic algorithms in search, optimization and machine learning. In *Proceedings of the 2000 Congress on*.
Hoff, A., Peiro, J., Corberan, A., & Marti, R. (2017). Heuristics for the capacitated modular hub location problem. *Computers & Operations Research*, *86*, 94-109.
Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.
Horner, M. W., & O'Kelly, M. E. (2001). Embedding economies of scale concepts for hub network design. *Journal of Transport Geography*, *9*(4), 255-265.
Hosseini, S. B. (2013). *A Lagrangean Relaxation Approach for the Modular Hub Location Problem* (Doctoral dissertation, Concordia University).
Jaillet, P., Song, G., & Yu, G. (1996). Airline network design and hub location problems. Location science, 4(3), 195-212.
Kahag, M. R., Niaki, S. T. A., Seifbarghy, M., & Zabihi, S. (2019). Bi-objective optimization of multi-server intermodal hub-location-allocation problem in congested systems: modeling and solution. *Journal of Industrial Engineering International*, *15*(2), 221-248.
Karimi, B., & Bashiri, M. (2019, January). Multi-period modular capacitated hub location problem with a splittable flow of commodities and multimodal transportation system. In *2019 15th Iran International Industrial Engineering Conference (IIIEC)* (pp. 31-36). IEEE.
Khaleghi, A., & Eydi, A. (2023). Multi-period hub location problem considering polynomial time-dependent demand. *Computers & Operations Research*, *159*, 106357.
Majima, T., Takadma, K., Watanabe, D., & Katuhara, M. (2017). Generating hub-spoke network for public transportation: comparison between genetic algorithm and cuckoo search algorithm. In *Intelligent and Evolutionary Systems: The 20th Asia Pacific Symposium, IES 2016, Canberra, Australia, November 2016, Proceedings* (pp. 263-275). Springer International Publishing.
Mikić, M., Todosijević, R., & Urošević, D. (2019). Less is more: General variable neighborhood search for the capacitated modular hub location problem. *Computers & Operations Research*, *110*, 101-115.
Mirzaghafour, F. S. (2013). *Modular hub location problems* (Doctoral dissertation, Concordia University).
Mitchell, M. (1998). *An introduction to genetic algorithms*. MIT press.
Momayezi, F., Chaharsooghi, S. K., Sepehri, M. M., & Kashan, A. H. (2021). The capacitated modular single-allocation hub location problem with possibilities of hubs disruptions: modeling and a solution algorithm. *Operational Research*, *21*(1), 139-166.
Mrabti, N., Hamani, N., Boulaksil, Y., Gargouri, M. A., & Delahoche, L. (2022). A multi-objective optimization model for the problems of sustainable collaborative hub location and cost sharing. *Transportation Research Part E: Logistics and Transportation Review*, *164*, 102821.
O'Kelly, M. E., & Bryan, D. L. (1998). Hub location with flow economies of scale. *Transportation research part B: Methodological*, *32*(8), 605-616.
O'kelly, M. E. (1987). A quadratic integer program for the location of interacting hub facilities. *European* journal of operational research, 32(3), 393-404.
O'Kelly, M. E. (1998). A geographer's analysis of hub-and-spoke networks. Journal of transport Geography, 6(3), 171-186.
Peker, M., Kara, B. Y., Campbell, J. F., & Alumur, S. A. (2016). Spatial analysis of single allocation hub location problems. *Networks and Spatial Economics*, *16*, 1075-1101.
Rabbani, M., Mokhtarzadeh, M., & Manavizadeh, N. (2021). A constraint programming approach and a hybrid of genetic and K-means algorithms to solve the p-hub location-allocation problems. International Journal of Management Science and Engineering Management, 16(2), 123-133
Skorin-Kapov, D., Skorin-Kapov, J., & O'Kelly, M. (1996). Tight linear programming relaxations of uncapacitated p-hub median problems. *European journal of operational research*, *94*(3), 582-593.
Tanash, M., Contreras, I., & Vidyarthi, N. (2017). An exact algorithm for the modular hub location problem with single assignments. *Computers & Operations Research*, *85*, 32-44.

Topcuoglu, H., Corut, F., Ermis, M., & Yilmaz, G. (2005). Solving the uncapacitated hub location problem using genetic algorithms. *Computers & operations research*, *32*(4), 967-984.
Wu, Q., Sun, Z., Benlic, U., & Lu, Y. (2023). A parallel adaptive memory algorithm for the capacitated modular hub location problem. *Computers & Operations Research*, *153*, 106173.
Yaman, H. (2005). Polyhedral analysis for the uncapacitated hub location problem with modular arc capacities. *SIAM Journal on Discrete Mathematics*, *19*(2), 501-522.
Yaman, H. (2008). Star p-hub median problem with modular arc capacities. *Computers & Operations Research*, *35*(9), 3009-3019.
Yaman, H., & Carello, G. (2005). Solving the hub location problem with modular link capacities. *Computers & operations research*, *32*(12), 3227-3245.
Wang, C., Liu, Y., & Yang, G. (2023). Adaptive distributionally robust hub location and routing problem with a third-party logistics strategy. *Socio-Economic Planning Sciences*, *87*, 101563.