



HAL
open science

A MILP model and memetic algorithm for the Hub Location Routing problem with distinct collection and delivery tours

Xiao Yang, Nathalie Bostel, Pierre Dejax

► **To cite this version:**

Xiao Yang, Nathalie Bostel, Pierre Dejax. A MILP model and memetic algorithm for the Hub Location Routing problem with distinct collection and delivery tours. *Computers & Industrial Engineering*, 2019, 10.1016/j.cie.2019.05.038 . hal-02524875

HAL Id: hal-02524875

<https://hal.science/hal-02524875v1>

Submitted on 25 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Title page for COMPUTERS & INDUSTRIAL ENGINEERING

A MILP model and memetic algorithm for the Hub Location and Routing problem with distinct collection and delivery tours

Xiao Yang^a, Nathalie Bostel^b, Pierre Dejax^c

^aShanghai Customs College, Department of Customs Administration, Shanghai 201204,
China

^bUniversity of Nantes, CNRS, LS2N, F-44000 Nantes, France

^cIMT Atlantique, CNRS, LS2N, 44300 Nantes, France

Corresponding author:

Nathalie Bostel
University of Nantes, LS2N, CNRS
58 Rue Michel Ange, 44606 Saint Nazaire, France
Tel: +33 2 40 61 39 35
Email: nathalie.bostel@univ-nantes.fr

Declarations of Interest : None

Acknowledgements: The authors wish to thank Professor Marc Paquet, from the ETS of Montréal, for his helpful suggestions on this research.

A MILP model and memetic algorithm for the Hub Location and Routing problem with distinct collection and delivery tours

Abstract

In this paper, we study the capacitated single allocation Hub Location-Routing Problem (CSAHLRP) with independent collection and delivery processes. We focus on the design of less-than-truck load (LTL) transport networks for the transport of goods between many shippers and their clients using flow concentration hubs. We seek to integrate the hub location and vehicle routing strategies such that the location of the hubs, the allocation of supplier/client nodes to hubs, the design of routes between nodes allocated to the same hub, as well as the inter-hub freight transportations, can be determined efficiently. We propose a mixed integer linear programming model for the problem with the aim of minimizing the total fixed and variable costs. Computational experiments based on the Australian Post (AP) data set geographical network are conducted with the CPLEX solver. Furthermore, we propose a memetic algorithm (MA) to solve large problems. Computational results show that the exact method can find optimal solutions for small instances and good feasible solutions for some medium-sized tests. In addition to also solving small problems to optimality, the MA succeeds in finding high quality solutions for medium and large CSAHLRPs efficiently.

Keywords: Hub Location-Routing Problem, LTL transport, mixed integer programming model, memetic algorithm

1. Introduction

In this paper, we study the design of freight transport networks for less-than-truckload (LTL) shipments, when freight companies have to collect small freight or parcels from many different origins before distributing them to various destinations. This is a well known problem in freight transportation and logistics. Several applications are concerned, such as the delivery of small parcels for classical business or the e-commerce, the transportation of food products on pallets from producing factories to retail stores scattered all over a territory, or the collection and distribution of mail between postal service

facilities. Usually for LTL shipments, direct connection for origins to their destinations is not economical, mainly due to the waste of vehicle capacity. Shipping goods through flow concentration hubs is the most cost-effective option. It consists in combining demands from several origins, consolidating them at hub facilities, and then shipping them to their destinations, possibly through another hub. The collection and delivery processes can be performed directly between suppliers/clients and a designated hub or through local pick-up/delivery tours, depending on the characteristics of goods to be shipped. In the scientific literature, the corresponding network design problem is referred to as the Hub Location Problem (HLP) in the former case and the Hub Location and Routing Problem (HLRP) in the latter case, which is the subject of this paper. Furthermore, depending on specific circumstances, pick-ups and deliveries may be integrated within the same tours, such as for postal services, or handled through separated collection and delivery tours for logistical or scheduling reasons; for instance, it is the case for general freight forwarders. In the latter case, which we focus on in this paper, collections from shippers (i.e. suppliers or producers) may indeed be typically undertaken in the late afternoon, while deliveries to clients (i.e. retail stores) are usually done during early hours in the next morning.

Since terminal location and vehicle routing decisions are interrelated, the coordination of these two aspects offers the promise of more effective and economical decisions and avoids suboptimal solutions. However, it is a challenging problem to combine these two decisions. A wide range of research on how to simultaneously deal with terminal locations and planning of vehicle tours can be found and such problems are named the Location-Routing Problems (LRP). The researched HLRP also integrates these two decision problems. It belongs to the class of Many-to-Many Location-Routing Problem (MMLRP) introduced by [Nagy and Salhi \(1998\)](#), which is a variant of the LRP. It deals with terminal hub locations, planning of vehicle tours and determination of inter-hub flow exchanges to satisfy customer demands. As a comparison, for the standard LRP, no flows are exchanged between the facilities, whereas flow exchanges are to be determined in the HLRP.

When one wants to design a transport network involving the location of facilities as well as optimize the suppliers/clients nodes allocation and routes to these (unknown) facilities, which is the problem we consider, solving location and routing problems independently of one another may lead to highly suboptimal solutions (increase of the costs). Quoting [Laporte \(1988\)](#), “location and routing are intertwined decisions which must be modelled and often optimized simultaneously”. This fact has been quantified for the first time by [Salhi and Rand \(1989\)](#). Such integrated problems have been studied for many years already. See the early papers or reviews by [Balakrishnan et al. \(1987\)](#),

Laporte (1988), Nagy and Salhi (1998), or more recent reviews quoted in our State of Art (i.e. Prodhon and Prins (2014), Schneider and Drexl (2017)). These network design problems are therefore considered with a long term strategic approach. In this context, routes have to be understood as itineraries to handle an average demand, and they are meant to be adjusted over time on a daily basis.

Although the HLP, the LRP and the Vehicle Routing Problem (VRP) have been widely investigated, previous research in the area of the HLRP is limited and devoted to specific cases. There does not exist any “generic” formulation for the HLRP. The goal of this paper is to propose a model and an efficient solution technique to solve the deterministic, capacitated single allocation HLRP (CSAHLRP) for the LTL transport network where collections of freight from suppliers and deliveries to clients are organized through distinct tours. Applications of this problem lay in sectors like LTL goods distribution. Decisions for solving it involve the determination of the number and location of hub facilities, the allocation of suppliers/clients to hubs, as well as the design of collection and delivery routes and the inter-hub transportation flows. The overall goal is to find solutions minimizing the total fixed and variable costs. Computational experiments are conducted, based on a set of instances generated from the geographical Australian Post (AP) network and the parameters of the logistics and freight transport sectors. We compare solutions obtained by solving a MILP model using the CPLEX solver for small instances and a memetic algorithm, which is able to efficiently solve instances of small, medium and large sizes.

The paper is structured as follows. Section 2 gives a literature review concerning the HLRP and related problems. In Sections 3, a mixed integer linear programming model is presented for the target problem. Section 4 proposes a memetic algorithm (MA) for solving large instances of the problem. Computational experiments on both the exact and metaheuristic methods are described and analyzed in Section 5, proving the efficiency of our approach. Section 6 draws conclusions about this research and puts forward ideas for further study.

2. Related literature

As indicated above, the Hub Location-Routing Problem (HLRP) is closely related to the Hub Location Problem (HLP) and the Location-Routing Problem (LRP). For these two classes of problem we refer the reader to the relevant reviews. Campbell (1994) gave the first survey of the discrete HLP, Alumur and Kara (2008) provided a comprehensive review of the HLP covering the period from 1987 to 2007, Campbell and O’Kelly (2012)

presented the origins and motivations of 25 years of HLP research, and [Farahani et al. \(2013\)](#) highlighted the aspects of HLP published after 2007 and current trends.

Early surveys of the Location-Routing Problem can be found in [Balakrishnan et al. \(1987\)](#) and [Laporte \(1988\)](#). Recent reviews have also been carried out by, among others, [Nagy and Salhi \(2007\)](#), [Prodhon and Prins \(2014\)](#) and [Schneider and Drexl \(2017\)](#).

To our best knowledge, little research has been published on the HLRP, which is the subject of our work. The articles on the HLRP described in this section are listed and their main characteristics compared in [Table 1](#). The notations used are explained in [Table 2](#). These tables illustrate the wide variety of the characteristics of problems addressed and solution techniques used. Major specificities of these problems are illustrated by the columns of [Table 1](#), in terms of: hub characteristics (with or without capacity, number of hubs fixed or not); non hub nodes (suppliers/clients) allocation; type of routing constraints; solution method; application/data base; maximum problem size solved; type of pick-up and delivery tours (simultaneous or distinct).

One can see from this description that the research problems may differ widely in their characteristics. There does not exist a single generic problem allowing a comparison. We first describe the problems for which the number of hubs to open and their location needs to be decided within the optimization.

As already mentioned, the HLRP was introduced for the first time by [Nagy and Salhi \(1998\)](#) under the name of Many-to-Many Location-Routing Problem (MMLRP). They considered the demands to be satisfied for each pair of customers (suppliers and clients) as well as capacity restrictions of terminals. The pick-up and delivery processes might be simultaneous or distinct but local tours were subjected to a routing length constraint. A single instance of 249 nodes was solved using a hierarchical heuristic. [Wasner and Zäpfel \(2004\)](#) addressed the Austria parcel service problem as a case study and investigated the possibility of direct connections between non-hub pairs for the capacitated single allocation MMLRP. Routes can combine deliveries and pick-ups in sequence. They developed a hierarchical heuristic method due to the difficulty in solving the problem by standard solvers. Only one small instance of 10 nodes was solved. [Catanzaro et al. \(2011\)](#) proposed a Partitioning-Hub-Location-Routing Problem (PHLRP), partitioning the target network into several sub-networks. For each sub-network, hub locations, multiple allocations and traffic routings were scheduled with an LTL system. They developed an Integer Programming (IP) model solved with a Branch and Cut algorithm and explored some valid inequalities to strengthen it. Experimentations were made with randomly generated instances of up to 450 nodes. [de Camargo et al. \(2013\)](#) provided a new formulation of the MMLRP. In their assumptions, the hubs and vehicles were subjected to

a fixed cost and each customer had to be visited once and was subjected to a charge, called the handling cost, for each assignment. Furthermore, they allowed pick-ups and deliveries to occur simultaneously and imposed a maximum tour duration but no capacity on vehicles. Their formulation led to a decomposition into two sub-problems, handled by an efficient Bender’s decomposition algorithm. Computational experiments were carried out based on the Australian Post (AP) standard data set. The approach was compared to solving the MILP model with CPLEX, and succeeded in solving problems with up to 100 nodes. In the study of [Setak et al. \(2013\)](#), a comprehensive MILP model of the HLRP was proposed. The aim was to build a general hub network topology to minimize the total cost, establishing hub nodes and inter-hub links, and connecting hub nodes and non-hub nodes. Paths containing only one hub node or a direct link from origin to destination were allowed. Their model was solved with the CPLEX solver. In their experiments, data from the Australian Post (AP) and Civil Aeronautics Board (CAB) were used for problem sizes of up to 20 nodes. [Mohammadi et al. \(2013\)](#) focused on a stochastic green HLRP (SGHLRP) and tried to minimize the total cost along with the environmental effect. A multi-objective MILP formulation was proposed and a Multi-Objective Invasive Weed Optimization algorithm (MOIWO) was applied to obtain Pareto optimal solutions. Computational results on randomly generated instances of up to 100 nodes showed that the proposed algorithm outperformed other multi-objective algorithms in the literature, such as the Non-dominated Sorting Genetic Algorithm-II (NSGA-II), the Pareto Archived Evolution Strategy (PAES) and the Strength Pareto Evolutionary Algorithm (SPEA). [Mokhtari and Abbasi \(2014\)](#) used a Variable Neighbourhood Particle Swarm Optimization Algorithm (VNPSO) to solve the many-to-many HLRP with uncapacitated hubs. The efficiency of the proposed algorithm was compared to a Bender’s decomposition algorithm using numerous samples of up to 300 nodes created randomly. Numerical results indicated that it performed better for large problems. [Bostel et al. \(2015\)](#) studied the HLRP applied to the special features of postal service systems in which collections and deliveries may occur simultaneously within the same routes. A MILP model was solved using CPLEX and a memetic algorithm based on a genetic algorithm and a local search was developed to solve the problem for problems instances from the AP data set of up to 100 nodes.

In the following, we now describe articles focusing on problems with predetermined open hub numbers (p hubs problems). Such problems are called Many-to-Many p Hub-Location Routing Problems (MM p HLRP).

[Çetiner et al. \(2010\)](#) studied hubbing and routing for postal delivery systems. They developed an iterative two-stage solution procedure for locating hubs and planning local tours. In their assumptions, they allowed multiple allocations of non-hub to hub nodes

and simultaneous pick-ups and deliveries, while imposing a maximum tour length constraint. They applied their procedure to a randomly-generated data set and studied a case from the Turkish postal service of 81 nodes. Sun (2013) decomposed the HLRP into two sub-problems: a HLP in the first stage and a multi-depot VRP in the second, allowing simultaneous pick-ups and deliveries. An Ant Colony Optimization (ACO) algorithm was applied to the two stages. They solved randomly generated instances of 200 nodes. Rieck et al. (2014) studied the Timbertrade industry and designed a many-to-many network structure with three layers: the supplier layer, the potential hub layer and the delivery layer. A mixed-integer linear model was proposed and applied to solve 15-node networks to optimality. For larger instances up to 140 nodes, the authors developed a multi-start procedure as well as a genetic algorithm. Rodríguez-Martín et al. (2014) introduced a Hub Location and Routing Problem that is very close to the Single Allocation p -Hub Median Problem (SA p HMP). This aimed to position p hubs, allocate customer nodes to a single hub and connect within the same tour customer nodes that have the same hub allocation. At most q nodes could be allocated to one hub and a single travelling tour was allowed for each hub. A branch-and-cut algorithm was developed to solve the problem on AP and CAB instances of 50 nodes. More recently, a solution method based on an Endosymbiotic Evolutionary Algorithm (EEA) was proposed by Sun (2015) to deal with the hub location and vehicle routing problems, with distinct pick-ups and deliveries. They solved randomly generated instances of 200 nodes. Zameni and Razmi (2015) proposed a mixed-integer formulation for a multimodal p -hub location-routing problem with simultaneous pick-ups and deliveries. They considered a multimodal transportation system with different characteristics. The model was solved with GAMS and a genetic algorithm was developed. They solved instances of 50 nodes from AP and CAB.

Our work, developed in the next sections, is devoted to the CSAHLRP with distinct pick-ups and deliveries for general many-to-many LTL transport of goods or parcels. We indeed assume that the size/weight of freight to be transported from suppliers and to clients is significantly smaller than the vehicle capacities to require vehicle routing collection or delivery processes, which is generally the case in retail distribution. We propose a new MILP model by integrating efficient flow variables and constraints of the VRP into the classic HLP formulations. Solutions are obtained by solving the model with the CPLEX solver, and developing a memetic algorithm. Instances adapted from the network of the AP data set have been used as well as logistics and cost parameters from professional data of freight transport. Instances up to 100 nodes have been solved.

As can be seen from this description and Table 1, our problem presents one or more specificities which differ from other works. This justifies the development of a specific

model and solution technique.

Table 1: Recent literature of the HLRP

Authors	Hub capacity	Number of hubs	Non-Hub allocation	Routing constraints	Solution method	Application /Data	Problem size	Pick-up /delivery
Our research	Yes	Unfixed	Single	Capacitated	CPLEX; MA ¹	AP	100	Distinct
Nagy and Salhi (1998)	Yes	Unfixed	Single	Length	Hierarchical	One instance	249	Simultaneous
Wasner and Zäpfel (2004)	Yes	Unfixed	Multiple + direct	Capacitated	Hierarchical	Austrian parcel	10	Simultaneous
Çetiner et al. (2010)	No	p hubs	Multiple	Length	Two-stage heuristic	Turkish postal	81	Simultaneous
Catanzaro et al. (2011)	No	Unfixed	Multiple	Node number	B&C ²	Random instances	450	Distinct
de Camargo et al. (2013)	No	Unfixed	Single	Time	BDA ³	AP ⁴	100	Simultaneous
Setak et al. (2013)	No	Unfixed	Single + direct	No	CPLEX	AP & CAB ⁵	20	Distinct
Sun (2013)	Yes	p hubs	Single	Capacitated	Two-stage ACO ⁹	Random instances	200	Simultaneous
Mohammadi et al. (2013)	Yes	Unfixed	Single	Capacitated; Arrival time	MOIWO ⁶	Random instances	100	Simultaneous
Mokhtari and Abbasi (2014)	No	Unfixed	Single	Time	VNPSO ⁷	Random instances	300	Simultaneous
Rodríguez-Martín et al. (2014)	No	p hubs	Single	Node number	B&C	AP & CAB	50	Distinct
Rieck et al. (2014)	No	p hubs	Single + direct	Capacitated	Multi-start + GA ⁸	Timbertrade industry	140	Distinct
Zameni and Razmi (2015)	No	p hubs	Single	Time	GAMS; GA	AP & CAB	50	Simultaneous
Sun (2015)	Yes	p hubs	Single	Capacitated	EEA ¹⁰	Random instances	200	Distinct
Bostel et al. (2015)	Yes	Unfixed	Single	No	CPLEX; MA	AP postal	100	Simultaneous

Table 2: Method notation

¹ MA	Memetic Algorithm	² B&C	Branch-and-Cut algorithm
³ BDA	Benders Decomposition Algorithm	⁴ AP	Australian Post standard data set
⁵ CAB	Civil Aeronautics Board data set	⁶ MOIWO	Multi-Objective Invasive Weed Optimization
⁷ VNPSO	Variable Neighborhood Particle Swarm Optimization algorithm	⁸ GA	Genetic Algorithm
⁹ ACO	Ant Colony Optimization algorithm	¹⁰ EEA	Endosymbiotic Evolutionary Algorithm

3. Problem definition and model formulation

The CSAHLRP studied here is defined on a complete directed graph $G = (N, A)$ containing a set of vertices N and a set of arcs A where $N = H \cup I \cup J$. $H = \{k \mid k = 1, 2, \dots, h\}$ presents a set of potential hubs. Each potential hub is capacitated and subjected to a fixed cost F_k once selected open. $I = \{i \mid i = h + 1, h + 2, \dots, h + m\}$ and $J = \{j \mid j = h + m + 1, h + m + 2, \dots, h + m + n\}$ stand for the sets of customers and clients who should be served. The numbers and locations of potential hubs, suppliers and clients are known data. Each pair of i and j ($i \in I, j \in J$) is associated with a given amount of freight flow q_{ij} to be shipped between them. The total supply $O_i = \sum_{j \in J} q_{ij}$ of supplier i and demand $D_j = \sum_{i \in I} q_{ij}$ of client j should be satisfied. The set $A = A_1 \cup A_2 \cup A_3$ includes the collection arc set $A_1 = \{(i, j) : i, j \in I \cup H\}$, the delivery arc set $A_2 = \{(i, j) : i, j \in J \cup H\}$ and the inter-hub transfer arc set $A_3 = \{(l, k) : l, k \in H\}$. A fleet of identical vehicles available for collections and deliveries is denoted as set V with a fixed capacity Q . Once a vehicle travels from non-hub node i to non-hub node j , a routing cost is incurred that is dependent on the distance d_{ij} . The transportation costs between two hubs are determined by travelling distances and transferred flows and inter-hub transport is not subject to capacity restrictions. To model the collection and delivery routes, a flow variable f_{ij} is used (see [Karaoglan and Altiparmak \(2015\)](#)), representing the vehicle load on each arc (i, j) . The following other constraints must hold: the capacity of each hub and collection/delivery vehicle must not be exceeded; the collection and delivery processes are independent; each route must start and end at the same selected hub; direct transportation for each supplier-client pair is forbidden; suppliers or clients on the same route must be assigned to only one selected hub; each vehicle can only be associated to one route.

The problem is to determine simultaneously the location of the hubs, the allocation of the suppliers and clients to hubs, the collection and delivery routing processes between nodes allocated to the same hub, as well as the inter-hub freight transportation. The optimization goal is to minimize the total fixed and variable costs. Long term strategic decisions costs (fixed costs for locating facilities) are thus considered together with operational decisions costs such as handling costs at facilities, interhub transport costs and routing costs between facilities and suppliers/clients considered on an average basis like the demand of goods to be shipped. Table (3) summarizes all the notations used in developing the corresponding HLRP model.

To model this problem, we propose a formulation of the HLRP containing binary variables with two indexes ($|N| \times |N|$ variables), and continuous variables with three indexes

($|H| \times |H| \times |I|$ variables), which reduces the size of the model compared to the formulations in other relevant articles (e.g., Mokhtari and Abbasi (2014), de Camargo et al. (2013)).

Table 3: Notation used in the model

<i>Sets</i>	<i>Description</i>
H	Set of hub nodes, $H = \{k \mid k = 1, 2, \dots, h\}$
I	Set of supplier nodes, $I = \{i \mid i = h + 1, h + 2, \dots, h + m\}$
J	Set of client nodes, $J = \{j \mid j = h + m + 1, h + m + 2, \dots, h + m + n\}$
N	Set of all nodes, $N = H \cup I \cup J$
A_1	Set of arcs in collection routing, $A_1 = \{(i, j) : i, j \in I \cup H\}$
A_2	Set of arcs in delivery routing, $A_2 = \{(i, j) : i, j \in J \cup H\}$
A_3	Set of arcs in inter-hub transfer, $A_3 = \{(l, k) : l, k \in H\}$
A	Set of all arcs, $A = \{(i, j) : i, j \in N\}$
<i>Parameters</i>	
F_k	Fixed cost of operating hub k
Γ_k	Capacity of hub k
C	Fixed cost of a vehicle
c_k	Handling cost to operate one unit product in hub k , $k \in H$
Q	Capacity of a vehicle
q_{ij}	Flow quantity from supplier $i \in I$ to client $j \in J$
d_{ij}	Distance between two nodes i and j , arc $(i, j) \in A$
cih	Unit cost parameter for the inter-hub transport (€/km.t)
cc	Unit cost parameter for the collection tour (€/km)
cd	Unit cost parameter for the delivery tour (€/km)
O_i	Total quantity of flow originating at supplier i , $O_i = \sum_{j \in J} q_{ij}$
<i>Decision variables</i>	
Y_{kl}^i	The fraction of flow shipped from supplier i via hubs k to l , $k, l \in H$, and $i \in I$
z_{ik}	The allocation variable of a node i to a hub k . It is equal to 1 if the node i is allocated to the hub k , 0 otherwise; especially, $z_{kk} = 1$ if the hub k is selected to be open, $i \in N, k \in H$
x_{ij}	Equal to 1 if a vehicle traverses arc (i, j) , and 0 otherwise
f_{ij}	Vehicle load on arc (i, j) if a vehicle travels directly from node i to node j , otherwise 0, $(i, j) \in A_1 \cup A_2$

Using the above notation, the HLRP model for minimizing total costs can be formulated as follows:

$$\begin{aligned}
\mathbf{M-HLRP} \quad & \min \sum_{k \in H} F_k z_{kk} + \sum_{i \in I} \sum_{(k,l) \in A_3} c_{ih} \cdot d_{kl} O_i Y_{kl}^i + \sum_{(i,j) \in A_1} cc \cdot d_{ij} x_{ij} \\
& + \sum_{(i,j) \in A_2} cd \cdot d_{ij} x_{ij} + \sum_{i \in I} \sum_{k \in H} c_k O_i z_{ik} + \sum_{i \in I} \sum_{(k,l) \in A_3} c_l O_i Y_{kl}^i + \sum_{k \in H} \sum_{i \in I \cup J} C x_{ki}
\end{aligned} \tag{1}$$

Subject to

— hub location constraints:

$$z_{ik} \leq z_{kk} \quad \forall i \in N, \forall k \in H \tag{2}$$

$$\sum_{k \in H} z_{ik} = 1 \quad \forall i \in I \cup J \tag{3}$$

$$\sum_{i \in I} O_i z_{ik} \leq \Gamma_k z_{kk} \quad \forall k \in H \tag{4}$$

$$\sum_{j \in J} D_j z_{jl} \leq \Gamma_l z_{ll} \quad \forall l \in H \tag{5}$$

$$\sum_{l \in H} Y_{kl}^i = z_{ik} \quad \forall i \in I, \forall k \in H \tag{6}$$

$$\sum_{k \in H} Y_{kl}^i O_i = \sum_{j \in J} q_{ij} z_{jl} \quad \forall i \in I, \forall l \in H \tag{7}$$

— collection routing constraints:

$$\sum_{j \in I \cup H} x_{ij} = 1 \quad \forall i \in I \tag{8}$$

$$\sum_{i \in I \cup H} x_{ij} - \sum_{i \in I \cup H} x_{ji} = 0 \quad \forall j \in I \cup H \tag{9}$$

$$x_{ki} \leq z_{ik} \quad \forall i \in I, k \in H \tag{10}$$

$$x_{ik} \leq z_{ik} \quad \forall i \in I, k \in H \tag{11}$$

$$x_{ij} + z_{ik} + z_{jl} \leq 2 \quad \forall i, j \in I, i \neq j, \forall k, l \in H, k \neq l \tag{12}$$

$$\sum_{j \in I \cup H} f_{ij} - \sum_{j \in I \cup H} f_{ji} = O_i \quad \forall i \in I \quad (13)$$

$$\sum_{i \in I} f_{ik} = \sum_{i \in I} z_{ik} O_i \quad \forall k \in H \quad (14)$$

$$f_{ij} \leq (Q - O_j) x_{ij} \quad \forall i \in I \cup H, \forall j \in I \quad (15)$$

$$O_i x_{ij} \leq f_{ij} \quad \forall i \in I, \forall j \in I \cup H \quad (16)$$

$$\sum_{i \in I} f_{ki} = 0 \quad \forall k \in H \quad (17)$$

$$f_{ik} \leq Q x_{ik} \quad \forall i \in I, k \in H \quad (18)$$

— domain of variables:

$$z_{ik} \in \{0, 1\} \quad \forall i \in N, \forall k \in H \quad (19)$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in N, \forall j \in N \quad (20)$$

$$0 \leq Y_{kl}^i \leq 1 \quad \forall i \in I, \forall k, l \in H \quad (21)$$

$$f_{ij} \geq 0 \quad \forall (i, j) \in A_1 \cup A_2 \quad (22)$$

— valid inequalities:

$$x_{ij} + x_{ji} \leq 1 \quad \forall i, j \in I \quad (23)$$

$$\sum_{k \in H} z_{kk} \geq \lceil \frac{\sum_{i \in I} \sum_{j \in J} q_{ij}}{\Gamma_k} \rceil \quad (24)$$

$$\sum_{k \in H} \sum_{i \in I} x_{ki} \geq \lceil \frac{\sum_{i \in I} O_i}{Q} \rceil \quad (25)$$

In addition, routing variables x_{ij} would be ignored in a preprocessing step whenever $O_i + O_j > Q, \forall i, j \in I, i \neq j$ or $D_i + D_j > Q, \forall i, j \in J, i \neq j$ (Karaoglan and Altiparmak (2015)).

The objective function (1) minimizes the total fixed and variable costs of the CSAHLRP network. More precisely, it includes the fixed cost for opening hubs, the transportation cost between hubs, local collection and delivery routing costs, the freight operating cost in hubs, and the fixed cost of routing vehicles once used. Constraints (2) ensure that non-hub nodes can be allocated to a hub only if the hub is open. Constraints (3) force each

non-hub node to be assigned to only one hub (single allocation). Hub capacity constraints (4) and (5) limit the total collection and delivery load on hubs. Constraints (6) and (7) are flow conservation equations. They impose the demand of each supplier or client to be served by the allocated hub (Ernst and Krishnamoorthy (1999)). Constraints (8) to (18) ensure a reasonable collection process. Indeed, constraints (8) guarantee that each supplier is visited just once. Constraints (9) guarantee an equal number of incoming and outgoing arcs. Constraints (10) - (12) eliminate illegal routes that do not start and end at the same hub. Constraints (13) are the flow conservation constraints for collections. Each time the vehicle serves a supplier, it must load all of its demand. Constraints (14) ensure that the total collection load entering each open hub equals the total demand of the suppliers who are allocated to the hub. Constraints (15) and (16) provide an upper and lower bound for the collection flows. Constraints (17) guarantee that the load on each vehicle is zero when leaving one open hub for the collecting process (Karaoglan and Altıparmak (2015), Yu and Lin (2016)). Constraints (18) impose that if there is no arc between a supplier node and a hub node, the flow should be zero. Constraints relative to the delivery processes represent conditions similar to those for the collection constraints ((8) to (18)). Constraints (19)-(22) specify the variables z_{ik} , x_{ij} , Y_{kl}^i and f_{ij} , respectively. Constraints (23) to (25) are valid inequalities. Constraint (23) are sub-tour elimination constraints for collections, which have been proposed for the LRP by Karaoglan et al. (2012) and are inspired from the classical sub-tour elimination constraints for the TSP (Dantzig et al. (1954)). Constraint (24) restricts the minimum number of open hubs considering that all the hubs have the same capacity. It was used by Karaoglan et al. (2012) and has been inspired from Belenguer et al. (2011). Finally, Constraint (25) provide a lower bound of the total number of vehicles required in any feasible solution. This type of inequality was first proposed by Achuthan et al. (2003) for a vehicle routing problem and adapted for the LRP by Karaoglan et al. (2012). Valid inequalities for delivery process are similar to constraints (23) and (25).

4. Memetic algorithm

As the HLRP integrates a hub location problem (HLP) and a vehicle routing problem (VRP), which are both NP-hard optimization problems, commercial solvers cannot solve large instances and a limited number of exact solution methods have been proposed. Thus, heuristic and metaheuristic algorithms have been developed in order to obtain good quality solutions in a reasonable computing time. Since the memetic algorithm (MA) has been proven efficient to solve similar problems such as the LRP and HLRP,

in this article, we propose a MA, combining a genetic algorithm (GA) and an iterated local search (ILS), to determine location and routing jointly.

The generic framework of the proposed MA is inspired from [Derbel et al. \(2012\)](#) and we have precisely adapted each step to the specificities of our problem: the initial population is generated using both heuristic methods to accelerate the convergence of the algorithm and randomized solutions to keep its diversity; each individual is evaluated by a fitness function corresponding to the objective function plus a penalty cost based on the capacity violation; pairs of parent solutions are selected by a unique fitness binary tournament selection, and a one-point crossover with a probability is applied to the selected parent pairs, followed by a mutation procedure (Section 4.4); next, local searches are implemented iteratively on the vehicle routing and hub location parts of the chromosomes to create new offspring (Section 4.5) and the current best solution is updated; finally, the newly-generated offspring is added to the current generation and the worst individual is eliminated. The whole process stops when the maximum number of iterations of the algorithm or the number of successive iterations without improvement is reached. The following sections describe the main phases of the MA in detail.

4.1. Solution representation and evaluation

A fundamental issue in designing a GA is to represent individuals as a set of chromosomes. Different genetic representations are proposed depending on the problem and they can affect the performance of the GA ([Ardjmand et al. \(2015\)](#), [Deng et al. \(2016\)](#)). In our approach, each chromosome $P(x)$ stands for one solution x and contains two vectors: the selected hub vector $H(x)$ and the non-hub nodes routing vector $A(x)$, according to the encoding scheme proposed by [Prins et al. \(2006\)](#). Vector $H(x)$ contains the selected hubs and their assignment configuration. Vector $A(x)$ records the permutation of suppliers and clients according to their sequence of service on a route. The positions of the open hubs in vector $H(x)$ are connected with their allocated suppliers and clients in vector $A(x)$. As one supplier/client must be served by only one hub, the vector $A(x)$ does not contain duplicates, that is, each supplier/client must appear only once in the vector $A(x)$. More specifically, consider a chromosome $P(x)$ and its two vectors $H(x) = \{h_1, h_2, \dots, h_n\}$ and $A(x) = \{a_1, a_2, \dots, a_n\}$. For every $i = \{1, 2, \dots, n\}$, if $a_i = j, h_i = k$, the non-hub node j is assigned to hub k . The tours for collections and deliveries can be deduced following the allocation scheme. According to the fixed sequence in vector $A(x)$, one vehicle starts from the first open hub and visits suppliers following the allocation scheme until the capacity of the vehicle is reached. Then for the same hub, another vehicle is used to start a new collection tour. The process contin-

ues until all the suppliers of all the hubs are routed. The same procedure takes place for the delivery routes. Figure 1 illustrates one representation of the chromosome $P(x)$ and its corresponding network sketch. This method of representation is simple and fast at simultaneously capturing locations and routings, which enables us to build feasible solutions and apply GA operators efficiently.

In order to compare and select the chromosome solutions, every individual is evaluated in terms of its fitness value, determined by a fitness evaluation function $F_{eva}(x)$. The fitness function $F_{eva}(x)$ of a solution x is defined as:

$$F_{eva}(x) = ObjValue(x) + Penalty(x) \quad (26)$$

$$Penalty(x) = \sigma \sum_{k \in H} \max\{0, \sum_{i \in I} z_{ik} O_i - \Gamma_k\} + \sigma \sum_{k \in H} \max\{0, \sum_{j \in J} z_{jk} D_j - \Gamma_k\} \quad (27)$$

where $ObjValue(x)$ denotes the objective value of solution x calculated by Equation (1) (see Section 3). σ is a penalty parameter with a large value, $\sum_{i \in I} z_{ik} O_i$ and $\sum_{j \in J} z_{jk} D_j$ represent the total demand of suppliers and clients allocated to an open hub k , Γ_k is the capacity of hub k , for loading or unloading operations. Thus the sum of the collection and delivery quantities violating the capacity of all open hubs is multiplied by the penalty parameter σ to calculate the penalty cost $Penalty(x)$ for solution x (Equation (27)).

4.2. Initialization of a population

To initialize the MA, an initial population must be generated. As mentioned in the global framework of the MA, the set of initial solutions is determined by heuristics as well as randomly. Some fast and simple heuristics guarantee a good quality of the initial generation while the randomly generated solutions ensure the diversity of the population.

More specifically, the minimum number $MinH$ of hubs to open is first calculated in order to satisfy the total demand. Then, the subsets H_1, H_2, \dots, H_n of the potential hub set H are generated. Each hub subset H_i must meet the minimum hub number requirement such as $MinH \leq |H_i| \leq |H|$. Each time, one of the hub subsets is selected non-repetitively as the current open hub configuration. Non-hub nodes are allocated to these open hubs by the allocation procedure. Under such an allocation scheme, two algorithms, the Nearest Neighborhood Algorithm (NN) and the Clarke and Wright Algorithm (CWA), are applied to create two different local collection and delivery routing

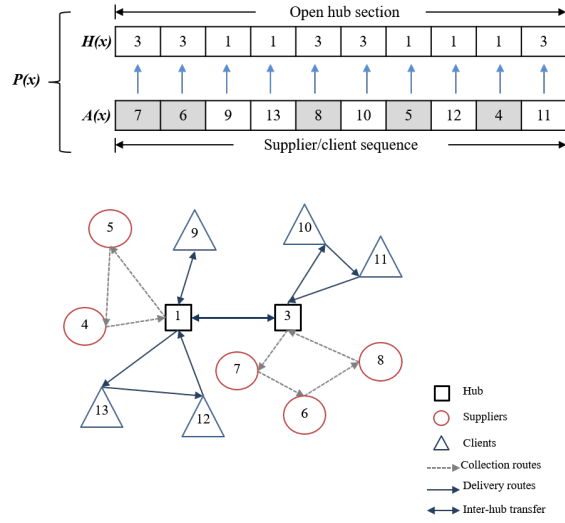


Figure 1: Representation of an HLRP solution

schemes. This means that two feasible solutions are generated at the same time with the same hub locations and allocations but different local tours. The process iterates until the predetermined population size is reached. Because of the limited number of hub subsets, the heuristic procedure may fail to reach the required population size. The remaining solutions are then randomly produced.

4.2.1. Allocation process

The allocation process is inspired from the allocation part of the Extended Clarke and Wright Algorithm (ECWA) proposed by [Karaoglan and Altiparmak \(2015\)](#). The main idea for allocating nodes to hubs is to determine for each non-hub node, the first and second nearest hubs and calculate the saving (difference in costs/distances) between the two allocations. Then, the non-hub nodes are sorted in non-increasing order of their savings. The non-hub nodes with the largest saving are allocated first, then the available hub capacity is updated and the process continues. When some nodes cannot be allocated to their nearest hub, their saving is recalculated. Furthermore, random new hubs may be opened if there are still suppliers/clients left because the remaining capacity of the open hubs is not enough to serve them. Hubs without customers are closed. The assignment procedures for collections and deliveries are independent.

4.2.2. Routing process

For each hub, one has to solve a routing problem for the non-hub nodes allocated to it. This routing problem is solved separately for suppliers and clients by two different classical heuristics : NN (Nearest Neighbour) and CWA (Clarke and Wright Algorithm). The NN approach is based on the simple idea of constructing routes progressively by inserting the nearest neighbour of the last inserted node into the route until the capacity of the vehicle is attained. New routes start from the hub until all the allocated non-hub nodes of a given hub are inserted.

An improved version of the CWA, proposed by [Caccetta et al. \(2013\)](#), generates the second routing solution. It calculates all the savings between non-hub nodes (difference in the cost attained when joining these two nodes in a route), and creates a savings list in a non-increasing order. The pairs of nodes are successively considered according to the savings list to build the routes. Three situations may occur: (i) if neither of the two nodes has been assigned to a route, a new route is built between them; (ii) if one of the two nodes has been included in an existing route without violating the vehicle capacity, the two nodes are connected and added to the same route; (iii) if both nodes have already been inserted into two different routes, the two nodes are not connected. The process repeats until all the nodes are inserted into a route.

4.3. Selecting parents for crossover

Since higher quality solutions can be captured with high-quality neighbourhoods, good parents should be selected for the crossover process to form the new generation. Different selection methods have been researched in the literature, such as tournament selection, roulette wheel selection, ranking selection and so on ([Reeves \(2003\)](#)). After preliminary experiments, we adopt a unique fitness binary tournament selection ([Fortin and Parizeau \(2013\)](#)) to keep the diversity of the selected parents.

The unique fitness binary tournament selection defines a unique fitness set \mathcal{F} , $\mathcal{F} = \{f_{eva}(i) | i \in |\mathcal{F}|\}$, including the unique fitness of the individuals from the current generation without repeating values. Here, $f_{eva}(i)$ stands for the value of the i th fitness in \mathcal{F} . If one fitness value is selected during the selection procedure, one of the solutions sharing the fitness value is randomly included in a selection set S . The advantage lies in limiting the chances of individuals with the same fitness to reproduce so that the preservation of diversity is stimulated.

4.4. Crossover and mutation

The selection procedure is followed by a one-point crossover operation on both the hub location vector $H(x)$ and the routing vector $A(x)$ with a probability P_c , simultaneously swapping nodes on selected parent pairs to form a new offspring. Two crossover points P_L and P_R are defined for the location part and the routing part, respectively. The hub location vector $H(x)$ of a new offspring combines the code of Parent 1 before crossover point P_L and the code of Parent 2 after P_L . On the routing vector $A(x)$, the new offspring takes the code of Parent 1 before P_R as the first part. The second part sequentially copies the code of $A(x)$ of Parent 2 except for the nodes that have been copied from Parent 1. Each time, two offspring are generated from each pair of selected parents.

The mutation operator mutates the chromosome by using two different methods on the two sections $A(x)$ and $H(x)$. In the location section $H(x)$, the hub assignment is modified by randomly choosing hubs to be replaced by others. Such a procedure makes it possible to open a new hub or close a hub. In the routing section $A(x)$, the random locations of two points are exchanged.

4.5. Local search method

The implemented local search (ILS) aims to improve further the newly generated offspring. Inspired by the method of [Manzour-al Ajdad et al. \(2012\)](#), first the local search is applied to the routing part of the chromosome with swap and insertion operators. Afterwards, four operators are applied sequentially on the hub section: hub replacement, hub closing, hub opening and hub swapping operators.

ILS on routing

The ILS procedure on the routing part of a selected chromosome uses the following two neighbourhoods:

- (1) Swap: Two non-hub nodes are selected to exchange their positions while other nodes remain unchanged.
- (2) Insertion: One non-hub node is shifted from its position and inserted into a random position in the routing vector. The operation is conducted on all the non-hub nodes. The insertion for each node is applied to all the positions to select the best one.

ILS on hub location and allocation

During the ILS on the hub location part, four local search neighbourhoods are used sequentially in order to better explore the hub location and non-hub allocation solutions.

- (1) Replacement. If not all the potential hubs have been selected to be opened in the hub location vector, one random open hub is chosen and replaced by a hub previously closed.
- (2) Closing. One open hub is randomly selected to be closed and is replaced by another currently open hub.
- (3) Opening. Several positions are randomly chosen and a new potential hub is opened and placed on the selected positions. If all the potential hubs are open, the operation will be skipped and other operations will continue.
- (4) Swap. Two positions are randomly selected and the corresponding hubs are exchanged while other nodes remain unchanged.

The two operators on the routing part of the chromosome provide two types of possible changes to the real routing schemes: changing the service sequence in the same route and conserving the same hub allocation; reassigning nodes to a new route, which alters not only the routing schedule but also the allocation scheme. On the other hand, the operators on the hub location part of the chromosome not only change the location and allocation of hubs but also change the routing schemes accordingly.

This way of designing the local searches can easily be implemented on the chromosome by just changing the positions of the nodes while altering the hub location, the allocation and local tour at the same time. This method revealed itself to be effective for our problem.

4.6. Illustration

As an example of our memetic procedure, we present a small example. The potential hub set comprises six nodes, numbered from 1 to 6, the supplier set 10 nodes numbered from 7 to 16 and the client set 10 nodes, numbered from 17 to 26 (some of the nodes are located at the same geographical location, others not). The hub capacities, as well as the vehicle capacities, are limited to 15 t. The total demand to be shipped from the suppliers to the clients is of 26,05 t. Figure 2 illustrates the best solution of the initial solution as well as the optimal solution of our memetic algorithm. Table 4 presents the results after the main algorithmic steps. One can observe in this small example that

the number of open hubs is stable (two), but open hubs change along the algorithmic steps, as well as non hub nodes allocations to hubs and routes. The objective function decreases regularly and significantly from 5122 to 4269, illustrating the efficiency of our procedure.

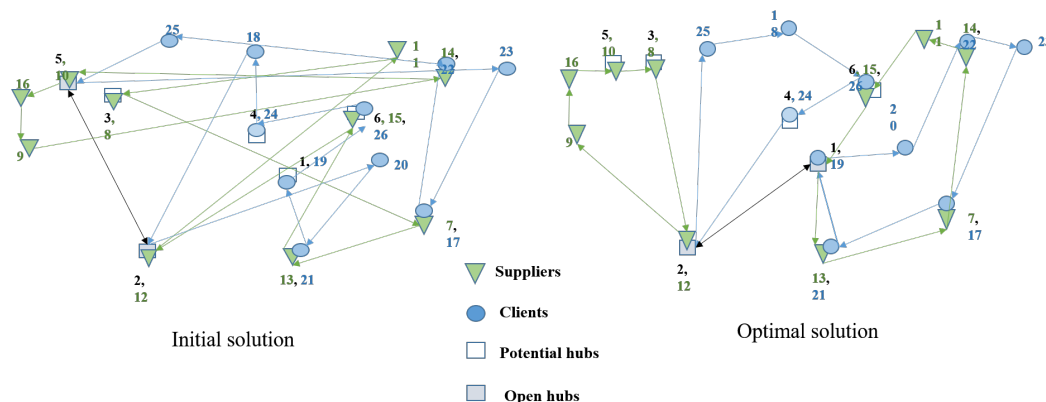


Figure 2: Illustrative example: initial and final solution

Table 4: Results of the memetic steps

Step	Open hubs	Total cost
Initial	2, 5	5122.69
Crossover & Mutation	2, 4	4524.46
Local Search	1, 4	4449.71
Optimal solution	1, 2	4269.15

5. Computational experiments

In this section, we first describe the data sets and parameters used for all the experiments in Section 5.1. Then, the implementation and assessment of the MILP model are presented in Section 5.2, where we discuss the tuning of CPLEX parameters and the efficiency of the valid inequalities of the MILP model. In Section 5.3, we describe the parameters and variants of the implementation of the memetic algorithm. Computational results are presented in Section 5.4, where we investigate the results obtained with CPLEX and the MA in terms of non-hub node allocation, hub location and collection/delivery routes depending on the sizes and parameters of instances. Lastly, some sensitivity analysis questions are discussed in Section 5.5 regarding the stability of the MA and the influence of fixed costs on the solutions.

The proposed MILP model is coded in Visual studio C++ 2012 and solved with CPLEX 12.6.1., while the MA approach is implemented in Visual studio C++ 2012 using PCs with 3.07 GHz and 8 GB RAM memory.

The general notation used in the tables is explained below:

- UB : best objective value found by CPLEX in 3 hours for each instance;
- LB : lower bound found by CPLEX within three hours;
- $\%Gap$: deviation in % between the best objective found by CPLEX and the lower bound found by CPLEX for each instance. Here, $\%Gap = \frac{UB - LB}{UB} \times 100\%$;
- Z_{best} : best objective value found by the MA in 10 runs for each instance;
- CPU time (s): total CPU time of CPLEX in seconds to obtain the best objective;
- $\%GapMA$: deviation in % between the best objective found by MA and the lower bound found by CPLEX for each instance. Here, $\%GapMA = \frac{Z_{best} - LB}{Z_{best}} \times 100\%$;
- T_{total} (s): total CPU time of 10 runs of the MA;
- Open hub: hub location scheme of the best solution;
- Route numbers: total number of collection and delivery routes of the best solution.

5.1. Data and parameters

Since there are no published benchmark instances corresponding to our problem, we have generated the problem instance networks on the basis of the geographic network of the Australian Post (AP) standard data set (Ernst and Krishnamoorthy (1999)). According to our hypothesis, it is assumed that the processes of picking up the freight from the suppliers and delivering it to the clients are distinct, and that the demand flows between each supplier-client pair are predetermined. Hubs may be located at the same geographical position as suppliers or clients or not. The distances between two nodes have been extracted from the AP data set. The flows (in tons) for each supplier-client pair have been generated uniformly within the interval $[0.15, 1.0]$ in order to be consistent with the capacity of the vehicles (15 tons, see Table 6) and the possibility of loading the freight of several suppliers (resp. clients) within the same collection (resp. delivery) routes.

We have generated 27 instances, each with up to 100 non-hub nodes and 10 potential hub nodes. Instances with 10 or 15 supplier and client nodes are referred to as “small instances”; those with 20 to 30 supplier and client nodes as “medium instances”, and those with 35 to 50 supplier and client nodes as “large instances”. In addition to different configurations of potential hub numbers, we consider three scenarios for each instance, with different hub capacities. By doing so, we can explore the influence of hub capacities on the solutions. The names of the instances are denoted as $|H|-|I|-|J|-\Gamma$. H stands for the candidate hub set, $|H| \in \{3, 6, 10\}$. I and J are sets of supplier and client nodes: $|I|=|J| \in \{10, 15, 20, 25, 30, 35, 40, 45, 50\}$. Γ is equal to the hub capacity. Capacities are integer numbers corresponding to $1/3$, $1/2$ or 1 of the total demand, depending on the instance size.

Tables 5 and 6 show the values of the parameters for hubs and vehicles. The parameters concerning the vehicles (fixed costs and unit transportation cost) are based on logistics data from the French Comité National Routier CNR¹ data base, the French road freight transport economic committee. The unit cost of handling unit flow in hubs was communicated by a French logistics company.

Table 5: The parameter values for hubs

Name	Value	Name	Value
Fixed cost F_k (€)	1000	Handling cost c_k (€/t)	1.78

Table 6: The cost parameter values for vehicles

Name	Value	Name	Value
Load capacity Q (ton)	15	Fixed cost for tour C (€)	100
Unit transfer cost c_{ih} (€/km.t)	0.057	Unit collection cost cc (€/km)	0.8
Unit delivery cost cd (€/km)	0.8		

5.2. CPLEX assessments

5.2.1. CPLEX parameter tuning

Since the solving strategy of CPLEX is controlled by a variety of parameters, tuning them for a given model and instance set is an efficient way of improving the performance of the solution process. Preliminary experiments were conducted using the tuning tool of CPLEX to analyze the HLRP model and determine the values of the parameters that

¹<http://www.cnr.fr/en>

might provide the best performance as a possible alternative to the default parameter setting. All the experiments of CPLEX in the article are conducted with a computing time limitation of 3 hours. In addition, to avoid a failure due to running out of memory for some instances, we set a tree memory limit of 1500 megabytes.

Experiments for tuning the CPLEX parameters with data sets of different sizes are conducted. The first and most critical CPLEX parameter tested is called “MIPEmphasis”, which controls the trade-offs between feasibility, optimality and speed in MILP solving. The second important CPLEX parameter, “Probe”, sets the extent of probing to be performed on variables before MILP branching. The results show that, in most cases, setting “MIPEmphasis” to 2 (emphasizing optimality over feasibility) yields competitive upper bounds and lower bounds with smaller gaps; the very aggressive probing level (value of “Probe” equal to 3) improved the solutions for most of the instances, especially by obtaining competitive lower bounds. These tuned CPLEX parameter values were retained in our further experiments. Note from the above tables that different pre-set CPLEX parameters may affect the solutions, generating different location, allocation and routing plans for the solutions of our problems.

5.2.2. Efficiency of valid inequalities in the MILP model

To analyze the efficiency of valid inequalities that we proposed in our MILP model, some tests were conducted to compare the results obtained with and without implementing these valid inequalities. Table 7 compares the results obtained for different sizes of instances. They show that including the valid inequalities in the proposed model provides a better performance in terms of improving lower bounds and decreasing gaps under the same time limitations (3 hours). It also decreases the upper bounds for most of the instance tests by up to more than 10%. The resulting solutions may consequently be changed in terms of opened hubs and the number of routes, which are generally reduced.

Table 7: Efficiency assessment of the valid inequalities

Instance $ H - I - J -\Gamma$	Model without valid inequalities					Model with valid inequalities				
	UB	LB	$\%Gap$	Open hub	Number of routes	$\%UB^1$	$\%LB^2$	$\%Gap$	Open hub	Number of routes
3-15-15-30	8828.59	8390.40	4.96	1, 2, 3	11	1.79	0.91	5.78	1, 2, 3	11
6-15-15-30	8707.18	7295.05	16.22	2, 4, 5	12	-3.04	5.16	9.14	1, 5, 6	13
10-15-15-30	8524.27	6680.63	21.63	5, 7, 10	12	1.32	7.89	16.55	1, 4, 5	12
3-20-20-45	10200.29	9593.82	5.95	1, 2, 3	18	-2.25	0.23	3.56	1, 2, 3	17
6-20-20-45	10435.48	8418.60	19.33	2, 3, 5	18	-0.74	7.30	12.79	3, 4, 5	17
10-20-20-45	10395.17	7901.05	23.99	2, 5, 10	19	-4.40	7.70	14.37	2, 8, 9	18
3-25-25-45	11811.05	10154.53	14.03	1, 2, 3	18	-2.62	-0.15	11.85	1, 2, 3	18
6-25-25-45	13502.91	8963.01	33.62	1, 2, 3	19	-10.13	4.18	23.05	1, 2, 3	17

Note: UB' and LB' are denoted as the upper bound and lower bound obtained by CPLEX with valid inequalities.

¹ Decreased/increased level of the upper bound (objective value): $\%UB = (UB' - UB)/UB \times 100\%$

² Decreased/increased level of the lower bound: $\%LB = (LB' - LB)/LB \times 100\%$

5.3. MA assessments

5.3.1. Parameter settings for the MA

The parameters of the proposed MA were tuned in terms of population size, maximum number of iterations and probabilities used for crossover and mutation. Preliminary experiments show that when the population size increases to 200, the overall performance of the MA is improved, but larger population sizes lead to no further improvement. Thus the size of the population for all the problem instances is set to 200. Furthermore, setting the probability of crossover and mutation to 0.8 and 0.7, respectively, is the best combination. In order to obtain high quality solutions, the MA was run 10 times for each instance (this policy will be evaluated later on in view of the experimental results). Below are some justifications to support these parameter settings.

We iterated the MA process up to 1000 iterations and recorded the results every 100 iterations for 10 runs each, i.e. for iteration number $\{100, 200, 300, \dots, 1000\}$. The results show that the solutions of small and medium instances remain unchanged after 100 iterations in all 10 runs of the MA. On the other hand, the MA can improve solutions of large instances until the iterations reach 200. For example, Figure 3 plots the average results of 10 runs for successive iterations of the MA for the large instance 10-40-40-75. The solutions are improved continuously until the generation of 140. More iterations lead to no more improvement.

Therefore, two stopping criteria were set for each run of the MA: the algorithm stopped if no improvement was obtained after 100 iterations or after a maximum of 200 iterations.

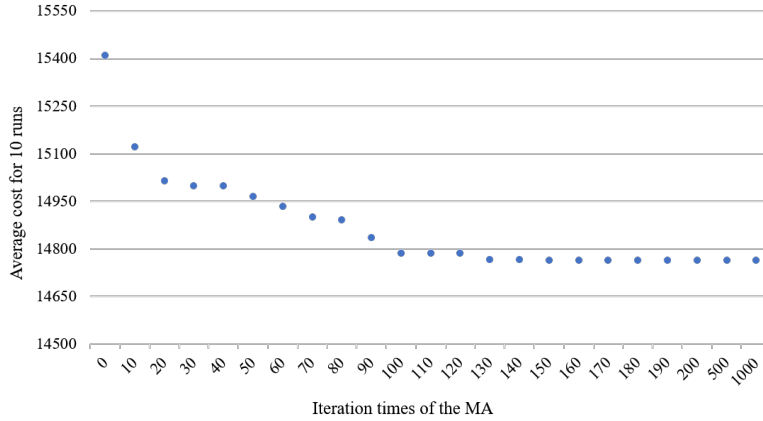


Figure 3: Average results of the MA in 10 runs up to 1000 iterations for instance 10-40-40-75

5.3.2. Implementation of the MA

In order to find the most efficient way to combine the operators of the MA for the allowed number of iterations (200), several variants were tested, especially the following two, denoted MA-1 and MA-2. Finally MA-2 was retained for our experiments with our complete set of instances.

For each iteration of the algorithm, MA-1 actually ignores the mutation process and executes systematically the full local search procedure on every newly generated offspring of the population after the crossover. This variant of the MA implementation yields good quality solutions but needs a very long computing time for large instances. In fact, the CPU running time of the LS phase consumes more than 95% of the total time of the algorithm. Alternatively, MA-2 fully executes the mutation step of the genetic part of the algorithm with a probability after the crossover process, but the LS is called only after 10 iterations of the genetic part, with the goal of reducing the computing time. Furthermore, the LS step is applied on three offspring only, which are randomly selected from the ten best offspring of the current generation. In addition, for each operator of the LS, once the target of the current offspring is improved, the procedure jumps to the next operator without considering the other offspring.

Table 8 presents the MA results of some small, medium and large instances run 10 times. Item %Time indicates the reduced percentage of the computing time when comparing MA-2 and MA-1. $\%Time = (T_2 - T_1) / T_1 \times 100\%$ where T_1 and T_2 are the CPU time of MA-1 and MA-2, respectively. Item %Cost indicates the increased or decreased percentage

of the total cost when comparing MA-2 to MA-1. $\%Cost=(Z_2 - Z_1)/Z_1 \times 100\%$ where Z_1 and Z_2 are the total cost of MA-1 and MA-2. For small and medium instances (e.g. instances 3-10-10, 3-25-25 and 6-30-30), the performance of MA-2 is comparable to that of MA-1 in terms of solution quality while MA-2 greatly reduces the total computing time compared to MA-1. For large instances (e.g. instances 10-45-45 and 10-50-50), the solutions found by MA-2 are slightly higher than those provided by MA-1 (from 0.45% to 3.72%). However, the computing time of MA-2 is reduced by almost 90% compared to MA-1. Thus, in order to achieve a good overall performance in terms of solution quality and computing time, we finally retained MA-2 for the experiments on our complete instance set. The results are presented in the next section.

Table 8: Computational tests on the two variants of the MA

Instance $ H - I - J $	Hub capacity	MA-1			MA-2				
		Z_{best}	CPU time(s)	Open hub	Z_{best}	CPU time(s)	Open hub	%Time	%Cost
3-10-10	10	5750.33	22.01	1, 2, 3	5746.53	16.52	1, 2, 3	-24.94%	-0.07%
	15	4269.15	23.49	1, 2	4269.15	12.60	1, 2	-46.36%	0.00
	30	3277.36	42.73	2	3277.36	14.36	2	-66.39%	0.00
3-25-25	45	11626.37	295.02	1, 2, 3	11676.80	83.38	1, 2, 3	-71.74%	0.40%
	60	10568.01	255.67	1, 3	10557.60	97.25	1, 3	-61.97%	-0.10%
	120	9938.67	372.607	1	9834.14	98.24	1	-73.64%	-1.05%
6-30-30	60	13114.97	1143.10	2, 4, 6	13052.10	306.87	4, 5, 6	-73.15%	-0.48%
	90	11976.11	1224.82	4, 6	11973.20	411.62	2, 6	-66.39%	-0.02%
	165	11408.26	1631.38	2	11401.80	338.96	2	-78.93%	-0.06%
10-45-45	75	13209.28	$\geq 4h$	3, 5, 9	13486.70	2630.10	3, 5, 9	-83.18%	2.10%
	105	12501.18	$\geq 5h$	8, 9	12557.00	2471.17	8, 9	-88.41%	0.45%
	195	12033.95	$\geq 8h$	8	12134.90	3025.74	1	-89.50%	0.84%
10-50-50	75	16284.53	$\geq 4h$	2, 3, 5, 10	16441.56	2953.06	2, 5, 10	-85.98%	0.96%
	120	15242.27	$\geq 8h$	2, 5	15471.72	5645.03	8, 10	-82.51%	1.51%
	225	14185.63	$\geq 13h$	8	14712.64	4960.14	8	-89.41%	3.72%

5.4. Analysis of computational results

Tables 9 and 10 present the problem solutions obtained by CPLEX and the MA. Table 9 reveals that the CPLEX solver can find optimal solutions for the smallest instances only. It encounters difficulties in solving medium size problems, obtaining low quality solutions with gaps of up to 39.91% (instance 10-30-30-165). The computing time with CPLEX reaches 3 hours for most of the instances, even small ones. The CPLEX solutions for large instances are not presented, as their gaps reach as high as 70%. The proposed MA finds solutions for all instance sets and is capable of solving some small problems to optimality. Moreover, for medium problems, it reduces the biggest gap of CPLEX from 39.91% to 13.64%. Most importantly, the MA dramatically reduces the computing time for the small and medium problems. Computing times for 10 runs for each instance lie

between 16 and 850 seconds for small- to medium-sized instances. In Table 10, Z_{aver} and Z_{best} represent the average and best objective values of 10 runs by the MA, respectively. Column $\%Z_{aver}$ records the average deviation with $\%Z_{aver} = (Z_{aver} - Z_{best})/Z_{best} \times 100\%$; column $\%Z_{max}$ records the deviation between the maximum and the minimum objective values in 10 runs and $\%Z_{max} = (Z_{max} - Z_{best})/Z_{best} \times 100\%$. The small gaps (up to 3.84% for $\%Z_{aver}$ and 7.69% for $\%Z_{max}$) for large instances prove the robustness and usefulness of the MA. Moreover, large instances are solved in no more than 6000 seconds for the largest of them (Instance 10-50-50-120). In terms of logistics solutions, the numbers of open hubs are usually identical for CPLEX and the MA but they may not correspond to the same hubs. The numbers of routes are comparable.

Figure 4 provides an insight into how the solutions are found by CPLEX and the MA within the optimization process for a small instance set (3-10-10-15) (the X axis has been limited to the first seconds of the optimization process). The time and relevant objective value of the MA refer here to the average values of the 10 runs. UB and LB indicate the upper bound and lower bound found by CPLEX. We observe that the MA can find the optimal solutions in a few seconds. Similar results are observed with larger instances. The initial solution found by the MA is much smaller than that found by CPLEX and the solution is improved continuously and quickly to obtain a good quality solution whose cost value is smaller than that of the solution found by CPLEX in 3 hours.

Figure 5 shows that the instance sets are more difficult to solve when the total number of nodes increases, and when the number of potential hubs increases. It also shows that for a given node number, increasing the hub capacity yields an equal or decreasing cost.

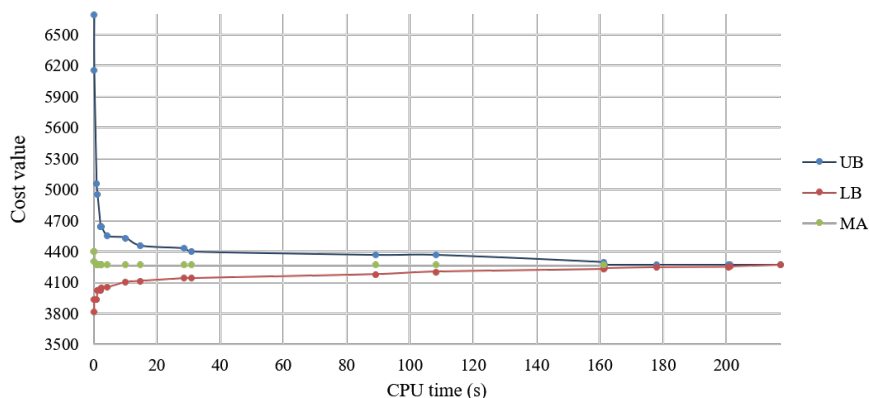


Figure 4: The evolution of solutions by CPLEX and the MA for instance 3-10-10-15

Table 9: Computational results for small- and medium-sized instances

Instance $ H - I - J $	Hub capacity	CPLEX						MA				
		UB	LB	$\%Gap$	CPU time(s)	Open hub	Number of routes	Z_{best}	$\%Gap$ MA	T_{total} (s)	Open hub	Number of routes
3-10-10	10	5666.52	5665.96	0.00	593.79	1, 2, 3	6	5746.53	1.40	16.52	1, 2, 3	6
	15	4269.15	4268.74	0.00	217.22	1, 2	4	4269.15	0.00	12.60	1, 2	4
	30	3277.36	3277.06	0.00	18.80	2	4	3277.36	0.00	14.36	2	4
6-10-10	10	5666.52	5379.10	5.07	10800.10	1, 2, 3	6	5659.78	4.96	58.57	3, 4, 6	6
	15	4269.15	4180.61	2.07	10800.00	1, 2	4	4269.15	2.07	59.04	1, 2	4
	30	3272.23	3271.92	0.00	157.76	4	4	3272.23	0.00	73.36	4	4
10-10-10	10	5792.80	5054.14	12.75	9192.72	1, 2, 8	6	5659.78	10.70	91.94	3, 4, 6	6
	15	4363.80	4169.43	4.45	10800.20	8, 10	4	4258.45	2.09	82.25	2, 10	4
	30	3245.62	3245.35	0.00	591.82	10	4	3245.62	0.00	87.76	10	4
3-15-15	30	8986.43	8467.11	5.78	10754.40	1, 2, 3	11	9015.79	6.09	22.89	1, 2, 3	11
	45	7584.04	7298.24	3.15	10536.69	1, 3	12	7638.35	4.45	34.19	1, 3	12
	90	6484.38	6279.90	3.15	10800.06	3	11	6539.41	3.97	60.63	3	11
6-15-15	30	8442.86	7671.57	9.14	10471.90	1, 5, 6	13	8127.86	5.61	32.68	1, 3, 5	12
	45	7107.65	6854.66	3.56	10800.15	3, 5	12	7107.65	3.56	40.88	3, 5	12
	90	6247.91	5977.35	4.33	10454.62	5	12	6179.78	3.28	89.62	5	11
10-15-15	30	8637.04	7207.55	16.55	10800.30	1, 4, 5	12	7863.46	8.34	114.30	5, 7, 10	12
	45	7013.07	6331.47	9.72	10800.11	7, 10	12	6861.88	7.73	131.45	7, 10	12
	90	6199.35	5974.59	3.63	10800.14	5	11	6199.35	3.63	145.86	5	11
3-20-20	45	9970.62	9615.48	3.56	10800.10	1, 2, 3	17	10024.00	4.08	65.34	1, 2, 3	16
	60	8825.23	8671.00	1.75	10800.09	2, 3	16	9048.58	4.17	82.64	2, 3	17
	120	8046.43	7749.60	3.69	10314.67	3	16	8057.46	3.82	83.86	3	16
6-20-20	45	10357.90	9033.56	12.79	10800.10	3, 4, 5	17	9806.10	7.88	113.05	2, 4, 5	17
	60	9257.22	8183.36	11.6	10800.23	4, 5	16	9022.21	9.30	116.44	3, 4	17
	120	8046.43	7684.09	4.50	10800.53	3	16	8041.56	4.45	128.07	3	16
10-20-20	45	9938.18	8509.61	14.37	10800.10	2, 8, 9	18	9632.53	11.66	280.64	8, 9, 10	18
	60	9520.10	7618.43	19.98	10800.30	5, 9	17	8725.87	12.69	244.04	7, 8	17
	120	7982.97	7017.70	12.09	10800.30	10	16	7885.86	11.01	257.96	7	16
3-25-25	45	11502.00	10139.00	11.85	9145.94	1, 2, 3	18	11676.80	13.17	83.38	1, 2, 3	18
	60	10602.79	9488.80	10.51	9369.65	1, 2, 3	18	10557.60	10.12	97.25	1, 3	16
	120	9865.98	9154.83	7.23	7024.90	1	16	9834.14	6.91	98.24	1	16
6-25-25	45	12135.20	9337.74	23.05	10800.30	1, 2, 3	17	11125.50	16.07	154.74	1, 2, 5	16
	60	12148.32	8963.21	26.22	10800.20	2, 5	19	10252.00	12.57	205.04	1, 5	16
	120	9036.62	8376.68	7.30	10800.09	5	16	9062.72	7.57	181.49	5	16
10-25-25	45	13209.30	8868.87	32.86	10800.20	1, 2, 3	18	10856.20	18.31	378.57	1, 8, 10	17
	60	13154.29	8043.80	38.85	10800.40	1, 3	24	10159.40	20.82	367.40	1, 8	16
	120	10281.30	7478.24	27.26	10800.19	5, 6	17	9056.17	17.42	406.98	5	16
3-30-30	60	15402.67	12812.91	16.82	10800.10	1, 2, 3	27	14732.00	13.03	124.47	1, 2, 3	25
	90	12945.41	11688.06	9.71	10800.20	1, 2	24	12994.00	10.05	164.00	1, 2	24
	165	11498.92	10928.82	4.96	10800.10	2	24	11401.80	4.15	147.97	2	23
6-30-30	60	14296.41	11439.38	19.98	10800.20	2, 4, 5	27	13052.10	12.36	306.87	4, 5, 6	24
	90	13838.82	10461.64	24.40	10800.90	2, 6	27	11973.20	12.62	411.62	2, 6	24
	165	12246.40	10531.57	14.00	10800.20	4	25	11401.80	7.63	338.96	2	23
10-30-30	60	17617.62	11161.86	36.64	10800.70	1, 5, 7	33	12952.60	13.83	653.84	2, 6, 7	23
	90	15289.92	10301.50	32.63	10800.70	4, 5, 8	27	11987.10	14.06	847.83	4, 6	22
	165	16362.83	9831.77	39.91	10800.90	1, 7, 8	34	11384.40	13.64	773.67	8	23

Table 10: Computational results of the MA for large-sized instances

Instance H - I - J	Hub capacity	Z_{best}	Z_{aver}	$\%Z_{aver}$	Z_{max}	$\%Z_{max}$	T_{total} (s)	Open hub	Number of routes
3-35-35	60	15303.30	15332.62	0.19	15389.20	0.56	97.10	1, 2, 3	27
	90	12704.90	12825.50	0.95	12938.80	1.84	178.36	1, 2	26
	180	11895.80	12096.61	1.69	12208.10	2.63	209.94	1	24
6-35-35	60	13486.80	13515.46	0.21	13597.20	0.82	364.61	1, 4, 6	27
	90	12747.70	12780.62	0.26	12798.70	0.40	301.76	1, 6	26
	180	11997.80	12114.40	0.97	12168.60	1.42	500.38	6	25
10-35-35	60	13579.80	13862.39	2.08	14318.20	5.44	552.54	1, 4, 6	27
	90	12655.50	12997.95	2.71	13498.50	6.66	1116.98	1, 2	26
	180	11975.80	12191.16	1.80	12682.20	5.90	1246.83	6	25
3-40-40	75	16172.60	16232.34	0.37	16315.00	0.88	200.44	1, 2, 3	30
	105	15233.90	15488.97	1.67	15914.20	4.47	295.92	2, 3	29
	210	13442.10	13615.45	1.29	13895.50	3.37	280.27	2	28
6-40-40	75	15257.80	15339.58	0.54	15391.70	0.88	324.14	3, 4, 5	31
	105	14317.30	14361.29	0.31	14442.70	0.88	507.60	4, 5	30
	210	13296.30	13395.88	0.75	13583.90	2.16	736.69	5	28
10-40-40	75	14586.30	14941.40	2.43	15630.40	7.16	1355.50	2, 7, 10	30
	105	13838.00	14369.58	3.84	14902.30	7.69	2591.37	2, 10	29
	210	13267.60	13507.67	1.81	14029.70	5.74	2021.55	4	28
3-45-45	75	14364.90	14485.52	0.84	14615.40	1.74	698.43	1, 2, 3	26
	105	13282.50	13399.95	0.88	13518.40	1.78	582.82	1, 3	25
	195	12165.40	12252.92	0.72	12370.70	1.69	483.29	1	25
6-45-45	75	13808.20	13964.20	1.13	14156.80	2.52	839.39	1, 2, 5	27
	105	12808.70	13234.49	3.32	13625.60	6.38	1153.75	1, 5	26
	195	12158.20	12261.14	0.85	12479.40	2.64	1067.82	1	25
10-45-45	75	13486.70	13838.82	2.61	14373.10	6.57	2630.10	3, 5, 9	28
	105	12557.00	12846.71	2.31	13224.00	5.31	2471.17	8, 9	26
	195	12134.90	12345.26	1.73	12598.30	3.82	3025.74	1	25
3-50-50	75	17242.60	17304.01	0.36	17387.10	0.84	219.578	1, 2, 3	33
	120	16324.10	16464.26	0.86	16539.80	1.32	723.49	1, 2	32
	225	14848.40	15007.79	1.07	15081.90	1.57	375.73	2	32
6-50-50	75	16669.30	16865.87	1.18	17003.30	2.00	2175.56	2, 3, 5	34
	120	15564.30	15733.77	1.09	15956.40	2.52	2558.65	2, 5	32
	225	14719.20	14967.48	1.69	15220.70	3.41	1231.40	2	30
10-50-50	75	16441.56	16811.56	2.25	17236.60	4.84	2953.06	2, 5, 10	33
	120	15471.72	15883.81	2.66	16496.90	6.63	5645.03	8, 10	32
	225	14712.64	15021.89	2.10	15842.10	7.68	4960.14	8	32

5.5. Sensitivity analysis

5.5.1. Stability assessment of the MA

In this section, we analyse the stability of the MA over the ten runs for the different instances. We computed the coefficient of relative standard deviation (RSD), $RSD = \frac{SD}{\bar{Z}} \times 100\%$, with ten runs. This indicator shows the variance in % of the objective values of ten runs compared to the average value. Here, SD is denoted as the standard deviation between the average objective value \bar{Z} in 10 runs and the best value Z_i found by the MA

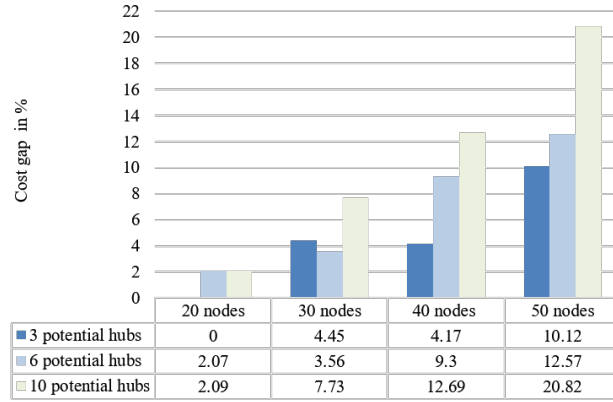


Figure 5: Solution gaps with different potential hub numbers obtained by the MA: the gap increases when the number of nodes and the number of potential hubs increase

for the i th run, $SD = \sqrt{\frac{\sum_{i=1}^{10} (\bar{Z} - Z_i)^2}{10}}$. Most of the RSD values are below 1%, and the average RSD is 0.86 % for all the instances. Figure 6 shows the average RSD value for different problem scales. Regarding the instances with different potential hub values, the average RSD values show relatively smooth curves without significant fluctuations. The stability of the MA proves its robustness and usefulness for decision-making. The small values of the RSD also suggest that just one or a few runs could be performed instead of ten to determine meaningful solutions of real-life problems in a dramatically reduced computing time. This would make it possible to test more alternative scenarios and sensitivity analysis, which is essential for decision-making.

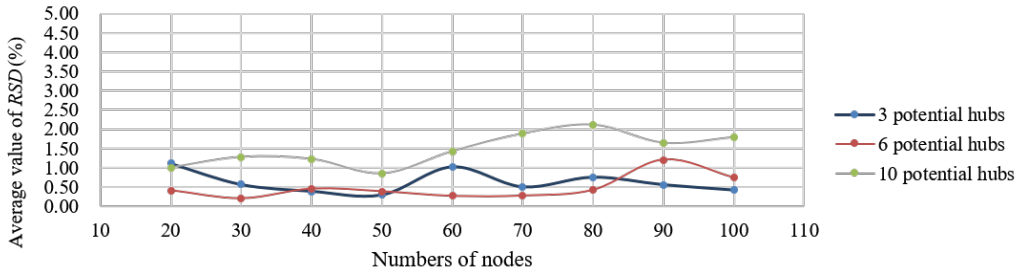


Figure 6: Average values of indicators with different numbers of instance nodes

5.5.2. Influence of the hub fixed cost

In the previous experiments, a value of 1000 was used for the hub fixed cost. However for real-life applications, the fixed cost of logistic facilities may be highly dependent on the particular case. In this section, we report a sensitivity analysis conducted on a possible range of fixed costs $F_k = \{1000, 500, 200, 100, 0\}$. All the results presented in Table 11 were obtained with the MA. In this table, the F_k column indicates the fixed cost value of opening one potential hub. The Fix_hub column represents the total fixed cost of the opened hubs. Column **Cost'** records the total cost of the solutions excluding the fixed hub cost, $\mathbf{Cost}' = Z_{best} - \text{Fix_hub}$. The following five columns detail **Cost'** and stand for the collection cost, the delivery cost, the inter-hub transfer cost, the handling cost and the fixed cost of using the vehicles, respectively. It can be observed in Table 11 that, for a given hub capacity, the fixed hub cost has no obvious influence on the solutions of small instances (e.g. instance 6-20-20) until it is set to a small value of 100 or even 0, which leads logically to opening more hubs and a reduced value of **Cost'**. For large instances (e.g. instance 10-35-35), when the fixed hub cost decreases, the number of open hubs and the inter-hub transfer cost increase while the collection cost, delivery cost and **Cost'** decrease. When there is no fixed hub cost, all the potential hubs are open and the best solution (the lowest **Cost'**) is obtained. This confirms the fact that a larger number of open hubs yields a lower value of the total cost (excluding fixed hub costs). The total cost on an HLRP network is reduced by increasing the volume of inter-hub transportation, which is a fundamental hypothesis of the Hub Location Problems. It can also be observed that, for most cases, no matter the hub fixed cost, some potential hub locations are always chosen, which is a robust feature for decision making in real life.

6. Conclusions

In this paper, we studied the deterministic Single Allocation Capacitated Hub Location Routing Problem (CSAHLRP) for the design of transport networks from many suppliers to many clients. We addressed the specific case where the collection of goods from suppliers and the deliveries to clients allocated to the same hub have to be organized within different routes for logistical or scheduling reasons. This is a situation frequently encountered in logistic systems for less-than-truckload (LTL) transport. Therefore, this problem is of theoretical and practical significance in order to determine feasible and efficient solutions of realistic problems in a reasonable computing time. The main contribution of our work is the proposition and validation of a MILP model and a memetic metaheuristic for the CSAHLRP with the problem characteristics described above. This

Table 11: Sensitivity analysis on the hub fixed cost

Instance $ H - I - J $	Hub capacity	F_k	Open hub	Z_{best}	Fix_hub	Cost'	Collect cost	Deliv_cost	Inter-hub cost	Handle cost	Fix_V
6-20-20	45	1000	2, 3, 5	9786.87	3000	6786.87	2527.13	1789.44	548.92	321.38	1600
		500	2, 3, 5	8286.87	1500	6786.87	2527.13	1789.44	548.92	321.38	1600
		200	2, 3, 5	7386.87	600	6786.87	2527.13	1789.44	548.92	321.38	1600
		100	2, 3, 4, 6	7094.27	400	6694.27	2431.23	1571.51	762.95	328.59	1600
	0	1,2,3,4,5,6	6593.05	0	6593.05	2380.97	1489.97	782.03	340.09	1600	
	60	1000	3, 4	9022.21	2000	7022.21	2745.16	1835.18	458.48	283.39	1700
		500	3, 4	8022.21	1000	7022.21	2745.16	1835.18	458.48	283.39	1700
		200	3, 4	7422.21	400	7022.21	2745.16	1835.18	458.48	283.39	1700
		100	3, 4, 6	7120.05	300	6820.05	2745.16	1659.98	528.57	286.35	1600
	0	1,2,3,4,5,6	6585.17	0	6585.17	2380.97	1493.44	767.86	342.90	1600	
	120	1000	3	8118.95	1000	7118.95	3014.13	2320.09	0.00	184.73	1600
		500	3	7618.95	500	7118.95	3014.13	2320.09	0.00	184.73	1600
200		3	7334.59	200	7118.95	3029.77	2320.09	0.00	184.73	1600	
100		2, 3, 6	7185.71	300	6885.71	2737.47	1700.26	558.20	289.78	1600	
0	1,2,3,4,5,6	6585.17	0	6585.17	2380.97	1493.44	767.86	342.90	1600		
10-35-35	60	1000	1, 4, 6	13579.80	3000	10579.80	3184.48	3073.48	1102.38	519.44	2700
		500	1, 2, 4, 6	12096.50	2000	10096.50	2901.26	2727.64	1127.09	540.52	2800
		200	1, 2, 4, 5, 6, 8	10791.40	1200	9691.40	2473.98	2367.77	1495.23	554.38	2700
		100	1, 2, 3, 4, 6, 8	10375.90	600	9775.90	2359.16	2265.70	1597.45	553.55	3000
	0	1, 2, 3, 4, 5, 6,7,8,9,10	9544.05	0	9544.05	1993.95	2166.77	1717.42	565.92	3100	
	90	1000	1, 2	12655.50	2000	10665.50	3664.28	3270.35	672.05	448.86	2600
		500	1, 2, 7	11690.00	1500	10190.00	2981.73	3063.52	1060.79	484.00	2600
		200	1,2,4,6,8	10767.40	1000	9767.40	2474.04	2419.48	1425.76	548.10	2900
		100	1,2,4,6,8	10228.00	500	9728.00	2481.83	2474.24	1424.43	547.46	2800
	0	1, 2, 4, 5, 6, 7, 8	9477.82	0	9477.82	2157.88	2237.03	1619.83	563.07	2900	
	180	1000	6	11975.80	1000	10975.80	4011.31	4162.94	0.00	301.55	2500
		500	2, 6	11579.80	1000	10571.80	3423.21	3304.53	803.78	448.33	2600
200		1,2,4,6,8	10748.70	1000	9748.70	2596.53	2387.85	1423.35	541.00	2800	
100		1, 2, 4, 5, 6, 8	10211.90	600	9611.90	2429.35	2305.50	1520.47	556.85	2800	
0	1,2,3,4,5, 6,7,8,9,10	9526.86	0	9526.86	1952.32	2103.72	1800.78	570.04	3100		

framework corresponds to a realistic LTL network design problem for the delivery of goods of small to medium sizes from a set of suppliers to their clients through concentration hubs and it is frequently used by LTL carriers.

Our MILP formulation of the problem combines recent flow formulations that are usually used in vehicle routing and location-routing problems. Our memetic algorithm combines a constructive heuristic for the generation of the initial population, a specifically designed genetic algorithm and several local searches on the location and routing parts of the problem.

For the purpose of experiments, we generated our own benchmark of 27 instances of different sizes and characteristics, since no appropriate benchmark corresponding to our applications was available and we compared the results obtained with the CPLEX solver and our memetic algorithm. In order to enhance the performance of the solution techniques, we did preliminary tuning of the main parameter of CPLEX as well as those of the memetic algorithm and we compared different implementations of the algorithm. We then conducted extensive comparative experiments to solve the problem and model with the CPLEX solver and with the memetic metaheuristic. The results show that CPLEX can solve the model to optimality only for small instances and finds feasible solutions for medium size instances with difficulty. The gap increases with the size of the instances. Computing times reach 3 hours even for medium-sized instances and convergence is slow. This prevents solving large realistic instances in this fashion. Alternatively, our memetic algorithm finds feasible solutions to the problem for all types of instances in a reasonable computing time and with limited gaps compared to the lower bounds of CPLEX. Convergence of the memetic algorithm to good solutions is fast and the solutions are very stable.

Decision making for real applications do not reduce to experimenting on a single data set. It involves evaluating and comparing a significant number of alternative scenarios and performing sensitivity analysis on significant parameters as well as assessing the robustness of the determined hub locations. Having to conduct a significant number of experiments to solve a real problem is a reason to develop an efficient solution technique capable to find realistic solutions of this complex strategic problem in a reasonable computing time. Conducting such an application was beyond the scope of this paper. However we performed some sensitivity analysis on key parameters, such as the fixed cost of opening hubs, potential hub numbers and capacities.

Our approach could also be adapted to solve different variations of the HLRP, such as the p-HLRP where the number of hubs to be selected is imposed, or the HLRP with simultaneous pick-ups and deliveries. With the growing concern for green logistics and

especially for the reduction of air pollution from transport, another research direction would consist in extending our models and algorithm to a bi-objective approach, minimizing both costs and GHG emissions from transport.

References

- Achuthan, N. R., Caccetta, L., and Hill, S. P. (2003). An improved branch-and-cut algorithm for the capacitated vehicle routing problem. *Transportation Science*, 37(2):153–169.
- Alumur, S. and Kara, B. Y. (2008). Network hub location problems: The state of the art. *European journal of operational research*, 190(1):1–21.
- Ardjmand, E., Weckman, G., Park, N., Taherkhani, P., and Singh, M. (2015). Applying genetic algorithm to a new location and routing model of hazardous materials. *International Journal of Production Research*, 53(3):916–928.
- Balakrishnan, A., Ward, J. E., and Wong, R. T. (1987). Integrated facility location and vehicle routing models: Recent work and future prospects. *American Journal of Mathematical and Management Sciences*, 7(1-2):35–61.
- Belenguer, J.-M., Benavent, E., Prins, C., Prodhon, C., and Calvo, R. W. (2011). A branch-and-cut method for the capacitated location-routing problem. *Computers & Operations Research*, 38(6):931–941.
- Bostel, N., Dejax, P., and Zhang, M. (2015). A model and a metaheuristic method for the hub location routing problem and application to postal services. In *Industrial Engineering and Systems Management (IESM), 2015 International Conference on*, pages 1383–1389. IEEE.
- Caccetta, L., Alameen, M., and Abdul-Niby, M. (2013). An improved clarke and wright algorithm to solve the capacitated vehicle routing problem. *Engineering, Technology & Applied Science Research*, 3(2):pp–413.
- Campbell, J. F. (1994). Integer programming formulations of discrete hub location problems. *European Journal of Operational Research*, 72(2):387–405.
- Campbell, J. F. and O’Kelly, M. E. (2012). Twenty-five years of hub location research. *Transportation Science*, 46(2):153–169.

- Catanzaro, D., Gourdin, E., Labbé, M., and Özsoy, F. A. (2011). A branch-and-cut algorithm for the partitioning-hub location-routing problem. *Computers & operations research*, 38(2):539–549.
- Çetiner, S., Sepil, C., and Süral, H. (2010). Hubbing and routing in postal delivery systems. *Annals of Operations research*, 181(1):109–124.
- Dantzig, G., Fulkerson, R., and Johnson, S. (1954). Solution of a large-scale traveling-salesman problem. *Journal of the operations research*, 2(4):393–410.
- de Camargo, R. S., de Miranda, G., and Løkketangen, A. (2013). A new formulation and an exact approach for the many-to-many hub location-routing problem. *Applied Mathematical Modelling*, 37(12):7465–7480.
- Deng, J., Wang, L., Wang, S.-y., and Zheng, X.-l. (2016). A competitive memetic algorithm for the distributed two-stage assembly flow-shop scheduling problem. *International Journal of Production Research*, 54(12):3561–3577.
- Derbel, H., Jarboui, B., Hanafi, S., and Chabchoub, H. (2012). Genetic algorithm with iterated local search for solving a location-routing problem. *Expert Systems with Applications*, 39(3):2865–2871.
- Ernst, A. T. and Krishnamoorthy, M. (1999). Solution algorithms for the capacitated single allocation hub location problem. *Annals of Operations Research*, 86:141–159.
- Farahani, R. Z., Hekmatfar, M., Arabani, A. B., and Nikbakhsh, E. (2013). Hub location problems: A review of models, classification, solution techniques, and applications. *Computers & Industrial Engineering*, 64(4):1096–1109.
- Fortin, F.-A. and Parizeau, M. (2013). Revisiting the NSGA-II crowding-distance computation. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pages 623–630. ACM.
- Karaoglan, I. and Altıparmak, F. (2015). A memetic algorithm for the capacitated location-routing problem with mixed backhauls. *Computers & Operations Research*, 55:200–216.
- Karaoglan, I., Altıparmak, F., Kara, I., and Dengiz, B. (2012). The location-routing problem with simultaneous pickup and delivery: Formulations and a heuristic approach. *Omega*, 40(4):465–477.

- Laporte, G. (1988). Location routing problems. In *B. Golden & A. Assad (Eds.), Vehicle routing: Methods and studies*, pages 163–198. Amsterdam: North-Holland.
- Manzour-al Ajdad, S., Torabi, S. A., and Salhi, S. (2012). A hierarchical algorithm for the planar single-facility location routing problem. *Computers & Operations Research*, 39(2):461–470.
- Mohammadi, M., Razmi, J., and Tavakkoli-Moghaddam, R. (2013). Multi-objective invasive weed optimization for stochastic green hub location routing problem with simultaneous pick-ups and deliveries. *Economic Computation & Economic Cybernetics Studies & Research*, 47(3).
- Mokhtari, N. and Abbasi, M. (2014). Applying VNPSO algorithm to solve the many-to-many hub location-routing problem in a large scale. *European Online Journal of Natural and Social Sciences*, 3(4 (s)):647.
- Nagy, G. and Salhi, S. (1998). The many-to-many location-routing problem. *Top*, 6(2):261–275.
- Nagy, G. and Salhi, S. (2007). Location-routing: Issues, models and methods. *European Journal of Operational Research*, 177(2):649–672.
- Prins, C., Prodhon, C., and Calvo, R. W. (2006). A memetic algorithm with population management (MA|PM) for the capacitated location-routing problem. *Lecture notes in computer science*, 3906:183–194.
- Prodhon, C. and Prins, C. (2014). A survey of recent research on location-routing problems. *European Journal of Operational Research*, 238(1):1–17.
- Reeves, C. (2003). Genetic algorithms. *Handbook of metaheuristics*, pages 55–82.
- Rieck, J., Ehrenberg, C., and Zimmermann, J. (2014). Many-to-many location-routing with inter-hub transport and multi-commodity pickup-and-delivery. *European Journal of Operational Research*, 236(3):863–878.
- Rodríguez-Martín, I., Salazar-González, J.-J., and Yaman, H. (2014). A branch-and-cut algorithm for the hub location and routing problem. *Computers & Operations Research*, 50:161–174.
- Salhi, S. and Rand, G. (1989). The effect of ignoring routes when locating depots. *European journal of operational research*, 39:150–156.

- Schneider, M. and Drexl, M. (2017). A survey of the standard location-routing problem. *Annals of Operations Research*, pages 1–26.
- Setak, M., Karimi, H., and Rastani, S. (2013). Designing incomplete hub location-routing network in urban transportation problem. *International Journal of Engineering-Transactions C: Aspects*, 26(9):997.
- Sun, J. U. (2013). An integrated hub location and multi-depot vehicle routing problem. In *Applied Mechanics and Materials*, volume 409, pages 1188–1192. Trans Tech Publ.
- Sun, J. U. (2015). An endosymbiotic evolutionary algorithm for the hub location-routing problem. *Mathematical Problems in Engineering*, 2015.
- Wasner, M. and Zäpfel, G. (2004). An integrated multi-depot hub-location vehicle routing model for network planning of parcel service. *International Journal of Production Economics*, 90(3):403–419.
- Yu, V. F. and Lin, S.-Y. (2016). Solving the location-routing problem with simultaneous pickup and delivery by simulated annealing. *International Journal of Production Research*, 54(2):526–549.
- Zameni, S. and Razmi, J. (2015). Multimodal transportation p-hub location routing problem with simultaneous pick-ups and deliveries. *Journal of Optimization in Industrial Engineering*, 8(17):11–20.