



# Bi-objective load balancing multiple allocation hub location: a compromise programming approach

Rahimeh Neamatian Monemi<sup>1,4,5</sup>  · Shahin Gelareh<sup>1,3</sup> · Anass Nagih<sup>1,2</sup> · Dylan Jones<sup>6</sup>

Published online: 28 January 2020  
© The Author(s) 2020

## Abstract

In this paper we address unbalanced spatial distribution of hub-level flows in an optimal hub-and-spoke network structure of median-type models. Our study is based on a rather general variant of the multiple allocation hub location problems with fixed setup costs for hub nodes and hub edges in both capacitated and uncapacitated variants wherein the number of hub nodes traversed along origin-destination pairs is not constrained to one or two as in the classical models.. From the perspective of an infrastructure owner, we want to make sure that there exists a choice of design for the hub-level sub-network (hubs and hub edges) that considers both objectives of minimizing cost of transportation and balancing spatial distribution of flow across the hub-level network. We propose a bi-objective (transportation cost and hub-level flow variance) mixed integer non-linear programming formulation and handle the bi-objective model via a compromise programming framework. We exploit the structure of the problem and propose a second-order conic reformulation of the model along with a very efficient matheuristics algorithm for larger size instances.

**Keywords** Hub-and-spoke network design · Load balancing · Traffic engineering · Matheuristic

## 1 Introduction

While hub-and-spoke operation has been exercised in practice (especially in telecommunications) for over a half decade now, in the last two decades (and in particular in transportation

---

✉ Rahimeh Neamatian Monemi  
rahimeh.monemi@univ-lorraine.fr; r.n.monemi@gmail.com

<sup>1</sup> Laboratoire de Conception, Optimisation et Modélisation des Systèmes, LCOMS EA 7306, Metz, France

<sup>2</sup> Université de Lorraine, 57000 Metz, France

<sup>3</sup> Département R&T, IUT de Béthune, Université d'Artois, 62000 Béthune, France

<sup>4</sup> Université de Lorraine, IUT de Metz, 57000 Metz, France

<sup>5</sup> CORMSIS, Southampton Business School, University of Southampton, Southampton, UK

<sup>6</sup> CORL, Department of Mathematics, University of Portsmouth, Portsmouth, UK

systems of the post-2008 crisis) it has received a very significant attention from researchers. The research directions have become widespread in alignment with the variety of contemporary logistics and transportation business models of the post-2008 era and the agenda of Industry 4.0 (which focuses on collaborative and cooperative aspects) in parallel with the emergence of big data and Internet of Things (IoT).

The hub-and-spoke structures aim at exploiting economies of scale resulting in minimized total costs, maximized utilization of transporters and improved/maintained service levels among other advantages. In such structures the flows among origin–destination (O–D) pairs are routed via some consolidation/redistribution/redirection centers, which are referred to as *hub nodes* and are connected among themselves using inter-hub connections known as *hub edges*. The inter-hub connections have special features such as higher performance (increased bandwidth in telecommunications, faster trains in public transport or higher capacity of mega-vessels in liner shipping) and a potential to exploit economies of scale. While in single allocation models every spoke node is allocated to one unique hub node, in the general form of multiple allocation scheme, there are no constraints on the number of hub nodes to which a spoke node can be allocated. Such networks can be seen as two nested networks: a hub-level network and a spoke-level one.

In classical models of hub location—irrespective of single allocation or multiple allocation—it is often assumed that the flows will only traverse one or two hub nodes along every O–D path (Hall 1989; O’Kelly and Lao 1991; O’Kelly 1987). This assumption leads to models with very limited applicability such as in air transport while in almost all other application areas are considered as less a realistic approximation of operative networks in practice.

Apart from this, a typical vulnerability of hub-and-spoke networks in general is that such structures encourage a sort of non-uniform concentration of volumes (stress) with high variance among different hub edges and across the hub-level network, which puts a lot of pressure on a small number of segments or entities in this sub-network. The hub nodes and hub edges in this small subset process/carry much higher volumes compared to other hub facilities.

This work has been inspired by the issues encountered by the authors in various consultancy projects in telecommunication, land and maritime transport wherein the single objective optimal hub-and-spoke network solutions were practically unrealistic in terms of variance of distribution of flow and could be a cause of inefficiencies in service design and asset management—given the level of abstraction required to model the network design problem. The approach proposed in this paper aims to generate practice-friendly outputs by providing a small number of alternative solutions that illustrate the trade-off between minimization of cost and load balancing.

Figures 1 and 2 shed more light on the aforementioned issue in the context of a multiple allocation hub location problem where O–D flows can traverse any number of hub nodes and hub edges based on the service design on the hub-level structure [see Gelareh and Nickel (2011)].

Figure 1a, b correspond to the flow and cost matrices, respectively, of an instance with  $n = 10$  generated using the techniques proposed in Ernst and Krishnamoorthy (1996). The flow matrix is not symmetric, as in Fig. 1a, and contains a node with a supply/demand that is significantly higher than others. However, the cost structure is indeed symmetric and in particular, the cost of sending one unit of flow between node 0 and nodes 7, 8 and 9 is relatively high. This can be due to several reasons such as natural and geographical barrier in land transport or even security and insurance measures in maritime transport. For the same reason, the cost of installing a hub edge between those nodes has been devised to reflect this feature.

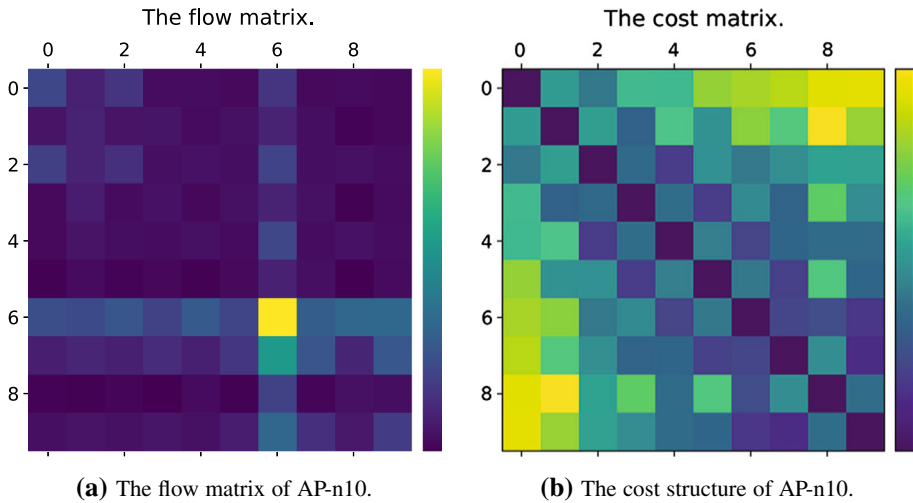


Fig. 1 The cost and flow structures of AP-n10 (derived from Australian Post dataset)

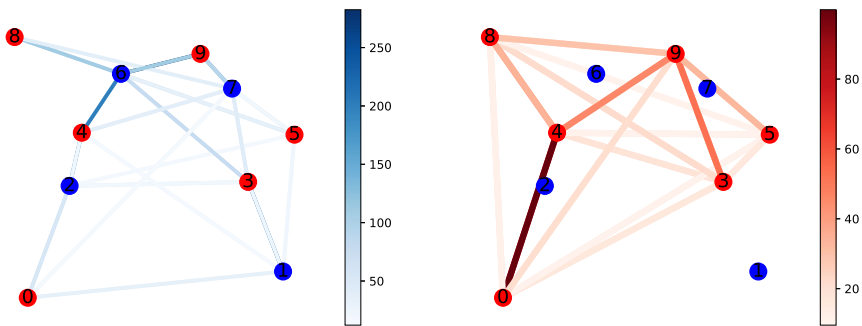


Fig. 2 The red circles correspond to the hub nodes and the edge colors are proportional to the flow density. (Color figure online)

Figure 2, depicts the optimal solution of an instance of a multiple allocation hub location problem [see Gelareh and Nickel (2011)]. The red nodes correspond to the hub nodes and the blue ones represent the spoke nodes. In the spoke-level network, in Fig. 2a, we observe a relatively balanced distribution of total flow on the spoke/access edges with the exception of the spoke edge 4–6. The latter is because as can be observed in the confusion map Fig. 1, the node 6 has an exceptionally high supply/demand in this instance, along with almost all the instances derived from the Australian Post (AP) dataset (Ernst and Krishnamoorthy 1996). This is the pattern for the multiple allocation scheme wherein every spoke node is allowed to send different portions of its flow via as many hub nodes as necessary.

However, in the hub-level network the volume of flow on hub edges is influenced by the way the spoke nodes split their receiving/sending flows among the hub nodes at which they are allocated. In Fig. 2b, one observes that the flow is not uniformly distributed across the network and certain edges and their end-points are receiving much more pressure (due to such a variance) compared to the other components. Such a pressure is a serious issue in practice,

especially if the end-nodes are in charge of some sort of processing, whether human factors are involved or sustainability measures play a role. In Fig. 2b, there is a highly saturated hub edge 0–4 and four other moderately exploited edges such as 4–8, 4–9, 3–9 and 8–9. The other hub edges are significantly under utilized.

A Decision Maker (DM) may not be convinced by such a solution and may think that the current structure imposes a lot of risks on their operation due to the saturation of 0–4. Moreover, they may argue that such a high pressure on this leg and the end points will significantly increase their costs of predictive and preventive maintenance while the other low utilized edges still require regular maintenance. In general, in this solution, the return on investment may hence be sub-optimal in practice. Our initial computational experiments show that this is a common phenomenon that median-type single objective hub location (hub-and-spoke network design) models are faced with.

The problem we are dealing with in this work is of a bi-objective nature as often two distinct stakeholder are involved each with its specific objective function—network owner willing to install elements on the network aiming at minimizing huge installation costs while satisfying the demands and network operators for whom the traffic management on the rented capacities is the vital part and they have not much to do with the huge strategic installation costs and investments. In some application areas such as telecommunications, two levels of network design are observed: Physical and Logical. The physical network design belongs to the infrastructure owners where they establish physical links (e.g. fiber optic cables etc.) and determine the location of hub nodes (e.g. routers, switching centers etc.). But the network operators do not make any significant investment on installing facilities and often only lease residual capacities. They mainly focus on the choice of a set of facilities and links from among existing ones and deploy their logical network. Here, the infrastructure owners are only in charge of establishing and maintaining the infrastructure and operators are in charge of efficiently routing the O–D demands. The former see themselves as the ones to provide strategically sufficient capacity ensuring the possibility for a balanced use of their infrastructures and the latter are in charge of avoiding congestion through different (congestion) pricing strategies at the tactical and operational levels.

## 1.1 Literature review

In the seminal work of Hakimi (1964), it has been shown that for finding the optimal location of a single switching center that minimizes the total wire length in a communication network, one can limit oneself to find the vertex median of the corresponding graph. Goldman (1969) generalized this result.

Assuming that every origin–destination (O–D) path traverses no more than two hub nodes dates back to the models in O’Kelly (1986a, b), which were the first continuous location models for the hub-and-spoke location problems dealing with locating two hub nodes in the plane and the discrete models proposed later by e.g. O’Kelly (1987). While this assumption is valid for the applications such as air passenger transport due to the aspects of service level, customer satisfaction, business models etc., it is rather hard to justify it in applications such as land transport, air cargo transport, maritime transport, public transport and many other sort of transport activities we observe in daily life.<sup>1</sup>

---

<sup>1</sup> It must be noted that visiting several hub nodes along an O–D path does not necessarily mean that loading/unloading (transshipment) will take place on every single intermediate hub node for a given demand. For instance, a number of containers originated from Tokyo and destined to Copenhagen, may be delivered to Shanghai as a major hub node and stay on-board a mega-vessel, which visits Singapore, Mumbai, Jebel Ali,

From the first round of works that started relaxing the classical assumptions of the Hub Location Problem, this topic has evolved significantly and more realistic models with better approximation of real practice have emerged. O’Kelly and Bryan (1998) dealt with flow-dependent costs and congestion cost was considered by de Camargo et al. (2009), De Camargo et al. (2011), de Camargo and Miranda (2012) among others. Rodríguez-Martín et al. (2014) proposed a model of hub location and routing; Correia et al. (2014) dealt with multi-product and capacity; and Sasaki et al. (2014) studied a competitive hub location based on Stackelberg games.

Very few multi-period hub location models have been proposed in the literature, that includes Campbell (1990), for a continuous approximation model of a general freight carrier serving a fixed region with an increasing density of demand, Gelareh (2008), Gelareh et al. (2015) for a HLP model within which the transport service provider starts with an initial configuration of the hub-level structure at the first period and then the network evolves in the course of a planning horizon and Contreras et al. (2011) for a dynamic uncapacitated hub location problem.

Very recently, Monemi et al. (2017) proposed a hub location problem in a co-opetitive setting. Interested readers are also referred to the recent HLPs literature review in Alumur and Kara (2008), Campbell and O’Kelly (2012) and Farahani et al. (2013) as well as other contributions by Campbell et al. (2002) and Kara and Taner (2011).

Concerning control of density of flow across the network, we can refer to the following works. Rahimi et al. (2016) dealt with a bi-objective model for a multi-modal hub location problem under uncertainty considering congestion in the hubs. Their objective functions attempt to minimize the total transportation cost as well as minimize the maximum transportation time between each pair of O–D in the network. Their proposed model is a nonlinear one and they resorted to metaheuristic approaches. De Camargo et al. (2011) dealt with a single allocation hub location problem under congestion, which is formulated as a nonlinear program and proposed a hybrid of outer approximation and a Benders approach to solve it. de Camargo and Miranda (2012) addressed a single allocation hub location problem under congestion and proposed a nonlinear formulation. de Camargo et al. (2009) studied the multiple allocation case with a nonlinear convex objective function. Elhedhli and Hu (2005a) dealt with a single allocation  $p$ -hub location problem and obtained an objective function with a non-linear cost term in it. They linearized the model, and provided an efficient Lagrangean heuristic. Elhedhli and Wu (2010) formulated congestion as a nonlinear mixed-integer program that explicitly minimizes congestion, capacity acquisition, and transportation costs. Congestion at hubs is modeled as the ratio of total flow to surplus capacity by viewing the hub-and-spoke system as a network of  $M/M/1$  queues. They used a Lagrangean heuristic on a linearized version of the nonlinear subproblems.

Table 1 summarizes the key features in some of the closely related contributions.

From the infrastructure owner perspective, a balanced distribution of flow across the hub-and-spoke network structure is a critical issue, which has received little attention in the literature (while congestion pricing from the operator viewpoints has received more attention). We study both *capacitated* and *uncapacitated* variants of the *Multiple Allocation Hub Location Problem* with fixed charge cost of hub nodes and edge. We consider two objective functions and deal with these in a compromise programming framework. To the best of our knowledge, this is the first approach of its type dealing with de-stressing in a hub-and-spoke structure in the literature and we are unaware of any similar work.

---

Footnote 1 continued

Suez, Hamburg and then be sent by a small vessel to Copenhagen without any actual transshipment taking place at Singapore, Mumbai, Jebel Ali or Suez.

**Table 1** A summary of main elements of the relevant contributions in literature

Work	Allocation scheme	No. hubs	Objective	Capacity	Congestion	Solution method
de Camargo and Miranda (2012)	Single	Endogenous	Min (cost + congestion)	None	Hub	Generalized benders
Elhedhli and Hu (2005b)	Single	Fixed	Min (cost + congestion)	None	Hub	Lagrangian heuristic
De Camargo et al. (2011)	Single	Endogenous	Min (cost + congestion)	None	Hub	Hyb. benders and outer approx.
Rahimi et al. (2016)	Single	Fixed	Duo (cost + time)	None	Hub	Metaheuristic
Elhedhli and Wu (2010)	Single	Fixed	Single (min)	Hub	Hub	Lagrangian relaxation
de Camargo et al. (2009)	Multiple	Endogenous	Single (min)	None	Hub	Hub generalized benders

## 1.2 Contribution and scope

This paper looks at the problem from the infrastructure owner's point of view and distinguishes between a load balancing (de-stressing) at the strategic level and congestion pricing in the lower levels. It is distinct from all other works by taking into account network design and hub-level flow balance (both at hub nodes and hub edges) in a bi-objective model. If we assume that once the infrastructure owner establish their facilities, the network operators will address their own problem of choosing among those hub nodes and hub edges to define their own sub-network and control their traffic by incorporating congestion cost. From the real-life application point of view, this work lies between these two levels. This paper presents the first generalized framework to deal with bi-objective models of CAPACITATED AND UNCAPACITATED MULTIPLE ALLOCATION HUB LOCATION PROBLEMS, which take into account load balancing in the strategic level. This work is inspired by some of the authors' consultancy projects with applications in liner shipping, land transport and traffic engineering in telecommunication applications. A compromise programming approach to the problem leads to non-linearities that are handled through perspective reformulation (leading to second order conic counterparts) and linearization for the capacitated and uncapacitated cases. In addition, by exploiting the structure of models and problems, we propose a very efficient matheuristic capable of producing effective high quality and practice-friendly solutions in reasonable time. We also validate our results and the efficiency of our approach on a real-life instance from the land transport industry.

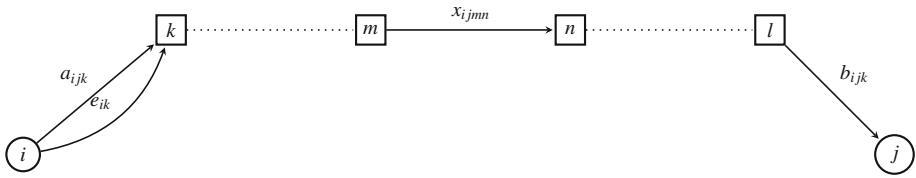
This paper is organized as follows: the mathematical models are proposed in Sect. 2, together with the techniques to transform the models to the more tractable counterparts. In Sect. 3, we describe our approach for dealing with the bi-objective model. Section 4 describes and elaborates on the design of our matheuristic. Section 5 reports some computational experiments and gives insights into the effectiveness of proposed approach. In Sect. 6, we summarize, draw conclusions and provide suggestions for further research directions.

## 2 Mathematical models

A Multiple Allocation Hub Location Problem which is referred to as HLPPT in Gelareh and Nickel (2011) constitute the starting point of our modeling. The variables and parameters in the proposed model are defined as in the following. It much be noted that for the ease of notation and without loss of generality we assume dealing with a complete graph, otherwise one can simply fix the variable corresponding to the non-existing elements to 0.

*Variables*  $x_{ijkl}$ ,  $\forall i, j \neq i, k, l \notin \{k, i\}$  represent the fraction of flow from  $i$  to  $j$  that traverses the hub edge  $\{k, l\}$ . Also,  $a_{ijk}$ ,  $\forall j \neq i, k \neq i, j$  stands for the fraction of flow from  $i$  to  $j$  that traverses spoke edge  $(i, k)$ , where  $i$  is not a hub and  $k$  is a hub node, and  $b_{ijk}$ ,  $\forall j \neq i, k \notin \{i, j\}$  represents the optimal path from  $i$  to  $j$  traversing  $(k, j)$ , where  $k$  is a hub node and  $j$  is not a hub. In addition,  $e_{ij}$ ,  $\forall i \neq j$  represents the fraction of flow from  $i$  to  $j$  that traverses  $(i, j)$  where at most one of  $i$  and  $j$  is a spoke node (see Fig. 3). For the hub-level variables,  $y_{kl} = 1$ ,  $k < l$  if the hub edge  $\{k, l\}$  is established, 0 otherwise and  $h_k = 1$  if node  $k$  is chosen to be a hub, 0 otherwise. We also define  $\dot{g}_{kl}(\cdot)$  and  $\dot{g}_k(\cdot)$  to represent the stress-related functions for the hub edge  $k - l$  and hub node  $k$ , respectively.

*Parameters* The transportation cost per unit of flow with origin  $i$  and destination  $j$  amounts to the sum of: (i) the cost of sending flow from  $i$  to the first hub node (if  $i$  is a spoke node assigned to a hub node) in the path to  $j$ , (ii) the cost incurred by traversing one or more hub



**Fig. 3** The correspondence between variables and network elements. The boxes represent hub nodes and circles represent spoke nodes

edges in the hub-level network discounted by the factor  $\alpha$ ,  $0 < \alpha < 1$ , and, (iii) the cost of transition on the last spoke edge (if destination is a spoke node). We introduce  $W_{ij}$  as the flow from  $i$  to  $j$ ,  $C_{ij}$  the cost per unit of flow on link  $i$  to  $j$ ,  $F_k$  fixed cost for establishing hub at node  $k$  and  $I_{kl}$  the fixed cost for establishing hub link between nodes  $k$  and  $l$ .

The De-stressing Hub Location Problem (DSHLP) in its general form follows: (DSHLP)

$$\begin{aligned} \min \quad Z_1 = & \sum_i \sum_{j \neq i} \sum_k \sum_{l \neq k} \alpha W_{ij} C_{kl} x_{ijkl} + \sum_i \sum_{j \neq i} \sum_{k \neq i, j} W_{ij} C_{ik} a_{ijk} + \\ & \sum_i \sum_{j \neq i} \sum_{k \neq i, j} W_{ij} C_{kj} b_{ijk} + \sum_i \sum_{j \neq i} W_{ij} C_{ij} e_{ij} + \\ & \sum_k F_k h_k + \sum_k \sum_{l > k} I_{kl} y_{kl} \end{aligned} \tag{1}$$

$$\min \quad Z_2 = \sum_k \sum_{l > k} \ddot{g}_{kl} + \sum_k \dot{g}_{kl} \tag{2}$$

s.t.

$$\sum_{l \neq i} x_{ijil} + \sum_{l \neq i, j} a_{ijl} + e_{ij} = 1, \quad i, j \neq i, \tag{3}$$

$$\sum_{l \neq j} x_{ijlj} + \sum_{l \neq i, j} b_{ijl} + e_{ij} = 1, \quad i, j \neq i, \tag{4}$$

$$\sum_{l \neq k, i} x_{ijkl} + b_{ijk} = \sum_{l \neq k, j} x_{ijlk} + a_{ijk}, \quad i, j \neq i, k \neq i, j, \tag{5}$$

$$y_{kl} \leq h_k, \quad k, l > k, \tag{6}$$

$$y_{kl} \leq h_l, \quad k, l > k, \tag{7}$$

$$x_{ijkl} + x_{ijlk} \leq y_{kl}, \quad i, j \neq i, k, l > k, \tag{8}$$

$$\sum_{l \neq i} x_{ijil} \leq h_i, \quad j, i \neq j, \tag{9}$$



$$\sum_{k \neq j} x_{ijkj} \leq h_j, \quad i, j \neq i, \quad (10)$$

$$e_{ij} \leq 2 - (h_i + h_j), \quad i, j \neq i, \quad (11)$$

$$a_{ijk} \leq 1 - h_i, \quad i, j \neq i, k \neq i, j, \quad (12)$$

$$b_{ijl} \leq 1 - h_j, \quad i, j \neq i, l \neq i, j, \quad (13)$$

$$a_{ijk} + \sum_{l \neq j, k} x_{ijkl} \leq h_k, \quad i, j \neq i, k \neq i, j, \quad (14)$$

$$b_{ijk} + \sum_{l \neq k, i} x_{ijkl} \leq h_k, \quad i, j \neq i, k \neq i, j, \quad (15)$$

$$e_{ij} + 2x_{ijij} + \sum_{l \neq j, i} x_{ijil} + \sum_{l \neq i, j} x_{ijlj} \leq h_i + h_j, \quad i, j \neq i, \quad (16)$$

$$\sum_{k, l} y_{kl} \geq 1, \quad (17)$$

$$\ddot{g}_{kl}(\cdot) \geq 0, \quad \forall k, l > k, \quad (18)$$

$$\dot{g}_k(\cdot) \geq 0, \quad \forall k, \quad (19)$$

$$x_{ijkl}, a_{ijk}, b_{ijk}, e_{ij} \in (0, 1), h_k, y_{kl} \in \{0, 1\}. \quad (20)$$

The first objective (1) accounts for the total cost of transportation plus hub node and hub edge setup costs while the second objective function (2) corresponds to the stress functions for hub nodes and hub edges. Constraints (3)–(5) are the flow conservation constraints. Constraints (6)–(7) ensure that both end-points of a hub edge are hub nodes. The constraints (8) ensure that a hub edge should exist before being used in any flow path. In (9) (resp. (10)) it is ensured that only a flow with origin (destination) of hub type is allowed to select a hub edge to leave the origin (arrive at the destination). Constraints (12)–(13) check the end-points of spoke edges. Any flow from  $i$  to  $j$  that enters to (departs from) a node other than  $i$  and  $j$ , must make sure that the node is a hub node. This is ensured by (14) ((15)). The choice of edges on the path between origin and destination ( $i$  and  $j$ ) depends on the status of  $i$  and  $j$ : whether both, none or just one of them is a hub. This is checked by (16). Constraint (17) ensure that there will be at least one hub edge in every feasible solution. Constraints (18) and (19) stand for the stress-related functions for the capacitated/uncapacitated versions.

## 2.1 Uncapacitated de-stressed fixed cardinality hub location problem (UDSFCHLP)

In the uncapacitated case, de-stressing is done by allocating the volume of flow in such a way that the variance of flow volume on the hub facilities is minimized.

The general form of such a load balancing objective function for the hub edges ( $\ddot{U}_{kl}$ ) follows (Altn et al. 2009):

$$\zeta \sum_{k,l>k} (\ddot{U}_{kl} - \ddot{U}^{mean})^2 + \beta \sum_{k,l>k} \ddot{U}_{kl}^2 \tag{21}$$

where  $\ddot{U}_{kl}$  represents the load of edge  $\{k, l\}$  and  $\ddot{U}^{mean}$  stands for the average load among the hub edges. The first term measures the sum of squared deviations from the mean over the hub edges and the second term controls the impact on the shortest path. As our first objective function already takes care of the transportation cost and distance, to avoid double counting, we set  $\beta = 0$  while  $\zeta = 1$ .

For the hub edges, the load is equivalent to the total flow transported on the hub edges:

$$\ddot{U}^{mean} = \frac{1}{|E|} \sum_{\{k,l\}} \ddot{U}_{kl} = \frac{1}{\sum_{k,l>k} y_{kl}} \sum_{k,l>k} \ddot{U}_{kl}, \tag{22}$$

$$\ddot{U}_{kl} = \sum_{i,j \neq i} w_{ij} x_{ijkl}, \quad \forall k, l > k, \tag{23}$$

and for the hub nodes, the load,  $\dot{U}_k$ , is equal to the flows arrived (sent or relayed) via adjacent hub nodes plus the ones received from the spoke nodes toward any destination other than the hub itself (using variables  $a_{ijk}$ ). The average load of hub nodes,  $\dot{U}^{mean}$ , is equivalent to the sum of flows arriving to all hub nodes divided by the total number of them.

$$\dot{U}^{mean} = \frac{1}{|H|} \sum_k \dot{U}_k = \frac{1}{\sum_k h_k} \sum_k \dot{U}_k, \tag{24}$$

$$\dot{U}_k = \sum_{i,j \neq i} \sum_{l \notin \{j,k\}} w_{ij} x_{ijlk} + \sum_{i,j \neq i} w_{ij} a_{ijk}, \quad \forall k. \tag{25}$$

The non-smooth optimization problems induced by constraints (22) and (24) is not efficiently handled by general-purpose black-box solvers such as CPLEX. We therefore, resort to a fixed cardinality approach and add the following two constraints to the formulation:

$$\sum_{k,l>k} y_{kl} = q. \tag{26}$$

$$\sum_k h_k = p. \tag{27}$$

Now, we define  $\ddot{g}_{kl}, \dot{g}_k \geq 0$  for the hub edge and hub nodes, respectively:

$$\ddot{g}_{kl} \geq \left( \sum_{i,j \neq i} w_{ij} x_{ijkl} - \frac{\sum_{k,l>k} \sum_{i,j \neq i} w_{ij} x_{ijkl}}{q} \right)^2, \quad \forall k, l > k, \tag{28}$$

$$\dot{g}_k \geq \left( \sum_{i,j \neq i} \sum_{l \notin \{j,k\}} w_{ij} x_{ijlk} + \sum_{i,j \neq i} w_{ij} a_{ijk} - \frac{\sum_k \left( \sum_{i,j \neq i} \sum_{l \notin \{j,k\}} w_{ij} x_{ijlk} + \sum_{i,j \neq i} w_{ij} a_{ijk} \right)}{p} \right)^2, \quad \forall k. \tag{29}$$

Therefore, the complete mathematical model for UDSFCHLP as follows:

(1)

(2)

s.t.

$$(3)–(20)$$

$$(26)–(27),$$

$$(28), \text{ and } (29).$$

Often the standard commercial general-purpose solvers have difficulties to handle this model in which case it can be reformulated as follows. Constraints (28) and (29) are quadratic and semi-positive definite, which (after introducing the required auxiliary variables  $\ddot{v}_{kl} \in \mathbb{R}$  and  $\ddot{f}_{kl} \in \mathbb{R}^+$ ) can be represented as a second order conic form as follows.

For hub edges, the reformulation yields:

$$\ddot{f}_{kl} \geq \frac{\sum_{k,l>k} \sum_{i,j \neq i} w_{ij} x_{ijkl}}{q} - \ddot{M}(1 - y_{kl}), \quad \forall k, l > k, \quad (30)$$

$$\ddot{f}_{kl} \leq \frac{\sum_{k,l>k} \sum_{i,j \neq i} w_{ij} x_{ijkl}}{q}, \quad \forall k, l > k, \quad (31)$$

$$\ddot{v}_{kl} = \ddot{f}_{kl} - \sum_{i,j \neq i} w_{ij} x_{ijkl}, \quad \forall k, l > k, \quad (32)$$

$$\ddot{g}_{kl} \geq (\ddot{v}_{kl})^2, \quad \forall k, l > k, \quad (33)$$

$$\ddot{v}_{kl} \in \mathbb{R}, \ddot{f}_{kl} \in \mathbb{R}^+, \forall k, l > k. \quad (34)$$

where  $\ddot{M}$  is a sufficiently large instance-dependent number no larger than  $\sum_{i,j \neq i} w_{ij}$ .

For the hub nodes we have:

$$\dot{f}_k \geq \frac{\sum_k \left( \sum_{i,j \neq i} \sum_{l \notin \{j,k\}} w_{ij} x_{ijlk} + \sum_{i,j \neq i} w_{ij} a_{ijk} \right)}{p} - \dot{M}(1 - h_k), \quad \forall k, \quad (35)$$

$$\dot{f}_k \leq \frac{\sum_k \left( \sum_{i,j \neq i} \sum_{l \notin \{j,k\}} w_{ij} x_{ijlk} + \sum_{i,j \neq i} w_{ij} a_{ijk} \right)}{p}, \quad \forall k, \quad (36)$$

$$\dot{v}_k = \dot{f}_k - \left( \sum_{i,j \neq i} \sum_{l \notin \{j,k\}} w_{ij} x_{ijlk} + \sum_{i,j \neq i} w_{ij} a_{ijk} \right), \quad \forall k, \quad (37)$$

$$\dot{g}_k \geq (\dot{v}_k)^2, \quad \forall k, \quad (38)$$

$$\dot{v}_k \in \mathbb{R}, \dot{f}_k \in \mathbb{R}^+, \forall k. \quad (39)$$

where  $\dot{M}$  is a sufficiently large instance-dependent number no larger than  $\sum_{i,j \neq i} w_{ij}$ .

It must be noted that for the (load balancing  $Z_2$ ) single objective UDSFCHLP, the optimal objective value is always 0 as, for any hub-level structure, it is always possible to route the flow such that the load on all the hub edges be equal.

### 2.2 Capacitated de-stressed hub location problem (CDSHLP)

In a capacitated environment, the congestion cost of a hub facility (edge or node) is proportional to the gap between volume of occupation and the nominal capacity of the network component. We introduce  $\Gamma_{kl}$  to be the capacity associated to the hub edge  $k - l$  and  $\Upsilon_k$  the capacity of hub nodes  $k$ . We assume that the congestion is an average delay of an  $M/M/1$  queue (as in Elhedhli and Wu 2010) and minimize the sum of convex delay cost functions associated with each hub facility.

We need to introduce a new set of capacity constraints to constrain the volume of flow that can be accommodated/processed by every hub facility (hub edge  $k - l$  and hub node  $k$ ).

$$\sum_{i,j \neq i} w_{ij}x_{ijkl} \leq \Gamma_{kl}y_{kl}, \quad \forall k, l > k. \tag{40}$$

$$\sum_{i,j \neq i} \sum_{l \notin \{j,k\}} w_{ij}x_{ijlk} + \sum_{i,j \neq i} w_{ij}a_{ijk} \leq \Upsilon_k h_k, \quad \forall k. \tag{41}$$

For hub edges, the stress function  $\ddot{g}_{kl}$  follows the following form:

$$\ddot{g}_{kl} \geq \frac{\ddot{U}_{kl}}{\left(1 - \frac{\sum_{k,l \neq k} \ddot{U}_{kl}}{\Gamma_{kl}}\right)} = \frac{\sum_{i,j \neq i} w_{ij}x_{ijkl}}{\left(1 - \frac{\sum_{i,j \neq i} w_{ij}x_{ijkl}}{\Gamma_{kl}}\right)}, \quad \forall k, l > k, \tag{42}$$

which can be written as:

$$\ddot{g}_{kl} \geq \frac{\Gamma_{kl} \sum_{i,j \neq i} w_{ij}x_{ijkl}}{\left(\Gamma_{kl} - \sum_{i,j \neq i} w_{ij}x_{ijkl}\right)}, \quad \forall k, l > k, \tag{43}$$

$$\ddot{g}_{kl} \left(\Gamma_{kl} - \sum_{i,j \neq i} w_{ij}x_{ijkl}\right) \geq \Gamma_{kl} \sum_{i,j \neq i} w_{ij}x_{ijkl}, \quad \forall k, l > k. \tag{44}$$

After some algebra, one obtains:

$$\left(\ddot{g}_{kl} - \sum_{i,j \neq i} w_{ij}x_{ijkl}\right)\left(\Gamma_{kl} - \sum_{i,j \neq i} w_{ij}x_{ijkl}\right) \geq \left(\sum_{i,j \neq i} w_{ij}x_{ijkl}\right)^2, \quad \forall k, l > k, \tag{45}$$

and as  $\ddot{g}_{kl} \geq \sum_{i,j \neq i} w_{ij}x_{ijkl}$  and  $\Gamma_{kl} \geq \sum_{i,j \neq i} w_{ij}x_{ijkl}$ , (45) is a rotated second order conic form.

In CDSHLP, if  $y_{kl} = 0$ , no flow is transported on this edge  $k - l$ , otherwise constraints (40) must be satisfied. Constraints (45) can therefore be replaced by the following rotated second order cone constraints:

$$\left(\ddot{g}_{kl} - \sum_{i,j \neq i} w_{ij}x_{ijkl}\right)\left(\Gamma_{kl}y_{ij} - \sum_{i,j \neq i} w_{ij}x_{ijkl}\right) \geq \left(\sum_{i,j \neq i} w_{ij}x_{ijkl}\right)^2, \quad \forall k, l > k, \tag{46}$$

given that  $\ddot{g}_{kl} \geq \sum_{i,j \neq i} w_{ij}x_{ijkl}$  and  $\Gamma_{kl}y_{ij} \geq \sum_{i,j \neq i} w_{ij}x_{ijkl}$ .

For the hub nodes, we have a similar development:

$$\begin{aligned} & \left(\dot{g}_k - \sum_{i,j \neq i} \sum_{l \notin \{j,k\}} w_{ij}x_{ijlk} - \sum_{i,j \neq i} w_{ij}a_{ijk}\right)\left(\Upsilon_k h_k \right. \\ & \left. - \sum_{i,j \neq i} \sum_{l \notin \{j,k\}} w_{ij}x_{ijlk} - \sum_{i,j \neq i} w_{ij}a_{ijk}\right) \geq \left(\sum_{i,j \neq i} \sum_{l \notin \{j,k\}} w_{ij}x_{ijlk} + \sum_{i,j \neq i} w_{ij}a_{ijk}\right)^2, \quad \forall k, \end{aligned} \tag{47}$$

The complete mathematical model for CDSHLP follows:

(1)

(2)

$$\begin{aligned}
 & s.t. \\
 & (3)–(20) \\
 & (46), \text{ and } (47).
 \end{aligned}$$

Again, most of the solvers have difficulties to deal with this form and this needs to be re-written by introducing some auxiliary variables.

For hub edges, we require  $\ddot{v}_{kl}^{(3)}, \ddot{v}_{kl}^{(4)}, \ddot{v}_{kl}^{(5)} \geq 0$ :

$$\ddot{v}_{kl}^{(3)} = \ddot{g}_{kl} - \sum_{i, j \neq i} w_{ij} x_{ijkl}, \quad \forall k, l > k, \tag{48}$$

$$\ddot{v}_{kl}^{(4)} = \Gamma_{kl} y_{ij} - \sum_{i, j \neq i} w_{ij} x_{ijkl}, \quad \forall k, l > k, \tag{49}$$

$$\ddot{v}_{kl}^{(5)} = \left( \sum_{i, j \neq i} w_{ij} x_{ijkl} \right), \quad \forall k, l > k, \tag{50}$$

$$\left( \ddot{v}_{kl}^{(5)} \right)^2 \leq v_{kl}^{(3)} v_{kl}^{(4)}, \quad \forall k, l > k, \tag{51}$$

$$\ddot{v}_{kl}^{(3)}, \ddot{v}_{kl}^{(4)}, \ddot{v}_{kl}^{(5)} \geq 0, \quad \forall k, l > k. \tag{52}$$

For the hub nodes, let  $\dot{f}_k = \left( \sum_{i, j \neq i} \sum_{l \notin \{j, k\}} w_{ij} x_{ijlk} + \sum_{i, j \neq i} w_{ij} a_{ijk} \right)$ , we have:

$$\dot{v}_k^{(3)} = \dot{g}_k - \dot{f}_k, \quad \forall k, \tag{53}$$

$$\dot{v}_k^{(4)} = \Upsilon_k h_k - \dot{f}_k, \quad \forall k, \tag{54}$$

$$\dot{v}_k^{(5)} = (\dot{f}_k), \quad \forall k, \tag{55}$$

$$\left( \dot{v}_k^{(5)} \right)^2 \leq \dot{v}_k^{(3)} \dot{v}_k^{(4)}, \quad \forall k, \tag{56}$$

$$\dot{v}_k^{(3)}, \dot{v}_k^{(4)}, \dot{v}_k^{(5)} \geq 0, \quad \forall k. \tag{57}$$

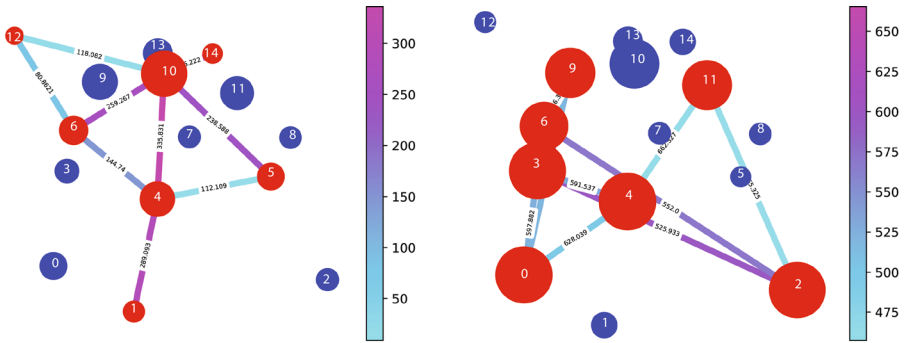
### 2.3 Illustrative examples and some reduction

In this section, we report the optimal solutions of four single objective problems corresponding to the capacitated and uncapacitated for cost and load balancing objective functions.

For the uncapacitated case, in Fig. 4, the optimal solutions of the single objective problems are depicted. The red circles represent the hub nodes and the blue ones stand for the spoke nodes. The node sizes (the diameter of nodes) are proportional to the volume of flow they process (supply, demand and transshipment) and the edge colors are proportional to the volume of flow on they carry (as shown by the color-map bar in each figure).

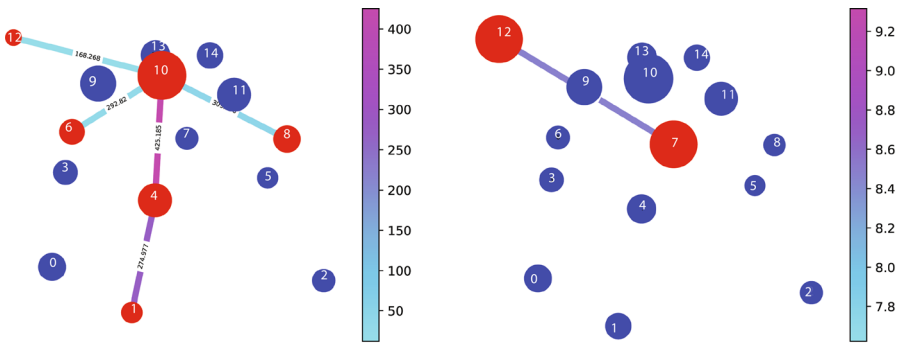
Expectedly, the network structures (in Fig. 4a, b) are quite different while also sharing some elements. As can be seen in Fig. 4a, the edges carry significantly different volumes of flow. For example, the edges 10–12 and 1–4 are on the extreme ends of the scale. However, in Fig. 4b the edges carry almost similar volumes and stick around an average.

For the capacitated case, Fig. 5, we observe again a similar pattern—the network structures are very different. Moreover, as the hub edges are subject of load balancing (not the spoke edges) therefore, often a very few and even very often only a single hub edge appears in an optimal solution of the load balancing problem (e.g. Fig. 5b) as this solution is the one that minimizes a hub-edge load balancing objective function.



(a) Single cost objective function. (b) Single load balancing objective function.

Fig. 4 Optimal solution of AP instance  $n = 15$  with  $q = 9$ ,  $p = 7$  and  $\alpha = 0.75$  in an uncapacitated environment



(a) Single cost objective function. (b) Single load balancing objective function.

Fig. 5 Optimal solution of AP instance  $n = 15$  and  $\alpha = 0.75$  in a capacitated environment

An important observation out of an extensive preliminary computational experiments is the following, which is very helpful in reducing the computational burden.

**Observation 1** *In a single objective model of load balancing, the balancing on the hub edges implies (with a very high precision) a load balancing on the hub nodes as well (but not vice versa). Therefore, the node-related variable and constraints can be safely removed in favor of a better computational behavior while the solution quality remains balanced and practice-friendly. However, still there is a constraint on the number of hub nodes that can appear in an optimal solution. This number cannot be larger than  $\frac{n}{2}$ —i.e. at most half of the nodes can become a hub. Therefore the constraints (27) will be modified to:*

$$\sum_k h_k \leq \frac{n}{2}. \tag{58}$$

Therefore, based on this preliminary extensive computational experiments, we opt to remove all the node-related variables and stick to the edge-based ones.

### 3 Bi-objective optimization: compromise programming

There is a large body of literature on method for dealing with the multi-objective optimization problems including goal programming, Pareto set construction, and compromise programming. In the case of our problem, Compromise Programming seems to be a viable approach compared to the goal programming techniques (because no information about the goal target values is available) and Pareto frontier generation approach (due to the nonlinearity in both CDSHLP and UDSFCHLP). Moreover Romero et al. (1998) also showed that the CP method works well with the two objective functions.

CP was first introduced by (Zeleny 1973) and relies on the reasonable assumption that DMs choice of solution is biased towards proximity the ideal point (Romero and Rehman 2003). This is formalized by the choice of a solution that minimizes an LP distance metric of the difference between the achieved and ideal values of the objectives.

This approach relies on minimizing the scaled distance between the ideal and efficient solutions Jones and Tamiz (2010). The relationship between the aforementioned approaches to deal with the multi-objective optimization problems is elaborated in Romero et al. (1998). On the other hand, Ballestero (2007) regards CP method as the maximization of the decision maker's additive utility.

Some successful applications of the CP approach include: (Anton and Bielza 2003) for a road project selection in Madrid, Spain; (André et al. 2008) for an application in macro-economic policy making in a general equilibrium framework for a case-study of Spain; (Fattahi and Fayyaz 2010) for an urban water management problem; (Amiri et al. 2011) for a multi-objective portfolio problem; (Liberatore et al. 2014) for a optimizing recovery operations and distribution of emergency goods; (Kanellopoulos et al. 2015) for scenario assessment; and recently Irawan et al. (2017) applied this technique on a model for installation scheduling in offshore wind-farms.

The general formulation of a CP approach is expressed as follows:

$$\text{Min } Lp = \left( \sum_1^n \left| \widehat{w}_i \frac{Z_i(x) - Z_i^*}{Z_{i*} - Z_i^*} \right|^p \right)^{\frac{1}{p}} \quad (59)$$

wherein,  $p \in [1, \infty]$  represents the distance metric and  $n$  is the number of objective functions. The  $p = 1$  solution represents pure optimization of the objectives, whereas the  $p = \infty$  solution represents balance between the objectives. The intermediate solutions in the range  $p \in (1, \infty)$  therefore represent the transition from a pure optimization to a balanced solution. Moreover,  $Z_i^*$  represents the ideal value of objective  $i$ ,  $Z_{i*}$  stands for the anti-ideal value of objective  $i$ ,  $Z_i(x)$  represents the achieved values of the  $i$ 'th objective and  $\widehat{w}_i$  stands for the weight/importance of objective  $i$  relative to the others. An anti-ideal solution for the load balancing model corresponds to a solution that installs some hub nodes/edges and routes the flow through shortest path causing a huge volume of flow traversing a few hub-level edges and an anti-ideal solution for the cost minimization model corresponds to a solution that installs many hub edges with very low utilization resulting.

Although the problems at hand are mostly non-linear ones, we may be still consider the properties that are known for the linear cases: One may limit oneself to  $p \in \{1, \infty\}$ , which can be used to derive a good approximation of intermediate compromise set points given that the problem is a bi-objective one (i.e.  $n = 2$ ) [see Romero et al. (1998)]. The CP objective function of CDSHLP and UDSFCHLP for  $p \in \{1, \infty\}$  are developed in the following:

**Case I**  $p = 1$

$$Min \mathcal{L}_1 = \left( \sum_{i=1}^n \left| \widehat{w}_i \frac{Z_i(x) - Z_i^*}{Z_{i^*} - Z_i^*} \right| \right), \tag{60}$$

which is equivalent to following linear objective function (in the bi-objective case).

$$\begin{aligned} \min \quad & \left( \sum_i \sum_{j \neq i} \sum_k \sum_{l \neq k} \alpha W_{ij} C_{kl} x_{ijkl} + \sum_i \sum_{j \neq i} \sum_{k \neq i, j} W_{ij} C_{ik} a_{ijk} \right. \\ & + \sum_i \sum_{j \neq i} \sum_{k \neq i, j} W_{ij} C_{kj} b_{ijk} + \sum_i \sum_{j \neq i} W_{ij} C_{ij} e_{ij} \\ & \left. + \sum_k F_k h_k + \sum_k \sum_{l > k} I_{kl} y_{kl} \right) \frac{\widehat{w}_1}{|Z_{1^*} - Z_1^*|} + \left( \frac{\widehat{w}_2 \sum_{k, l > k} g_{kl}^a}{|Z_{2^*} - Z_2^*|} \right) + \alpha_1 + \alpha_2 \end{aligned} \tag{61}$$

where  $\alpha_i = -\frac{\widehat{w}_i Z_i^*}{Z_{i^*} - Z_i^*}, \forall i \in \{1, 2\}$ .

**Case II**  $p = \infty$  The objective function (59) minimizes the maximum deviation ( $\pi$ ) as follows:

$$\begin{aligned} Min \mathcal{L}_\infty = \min \quad & \pi \\ s.t. \quad & \widehat{w}_i \frac{|Z_i(x) - Z_i^*|}{|Z_{i^*} - Z_i^*|} \leq \pi \quad \forall i = 1, \dots, n \end{aligned} \tag{62}$$

which is equivalent (in the bi-objective case) to:

$$\min \quad \pi \tag{63}$$

$$\begin{aligned} & \left( \sum_i \sum_{j \neq i} \sum_k \sum_{l \neq k} \alpha W_{ij} C_{kl} x_{ijkl} + \sum_i \sum_{j \neq i} \sum_{k \neq i, j} W_{ij} C_{ik} a_{ijk} \right. \\ & + \sum_i \sum_{j \neq i} \sum_{k \neq i, j} W_{ij} C_{kj} b_{ijk} + \sum_i \sum_{j \neq i} W_{ij} C_{ij} e_{ij} \\ & \left. + \sum_k F_k h_k + \sum_k \sum_{l > k} I_{kl} \right) \frac{\widehat{w}_1}{|Z_{1^*} - Z_1^*|} + \alpha_1 \leq \pi \end{aligned} \tag{64}$$

$$\left( \frac{\widehat{w}_2 \sum_{k, l > k} g_{kl}^a}{|Z_{2^*} - Z_2^*|} \right) + \alpha_2 \leq \pi \tag{65}$$

Figure 6 depicts the compromise objective function values for our bi-objective load balancing multiple allocation hub location problem. *A* and *B* are optimal values of cost and stress minimization hub location problems, respectively. While *E* is the nadir or an anti-ideal point, node *F* is the idea point wherein both single objectives are at their optimal. The line segment connecting points *C* and *D* sets an upper envelop on all the compromise solutions, i.e.  $\overrightarrow{DC}$ , is a representation of the compromise set. Towards the end point *D* the load balance is improved while the cost function will deteriorate and as we move further towards *C*, the cost objective function will improve and the load balancing objective function will deteriorate. As expected, the point *C* provides a better balance between the two objectives than point *D*.



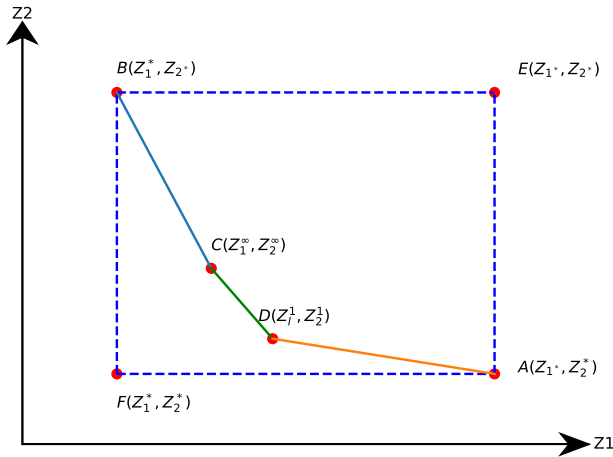


Fig. 6 Compromise solutions in the bi-objective load balancing multiple allocation hub location problem

### 4 Solution method

Compromise programming is an approach parameterized by weight factors  $\widehat{w}_i$ . In a transition from  $\mathcal{L}_1$  to  $\mathcal{L}_\infty$ , the computational behavior of the problem changes dramatically from a linear combination of objective functions to a minimax objective function, which is known to be computationally challenging Florentino et al. (2017). In addition, our initial computational experiments with different decomposition techniques revealed that different combinations of  $\widehat{w}_i$  in both  $\mathcal{L}_1$  and  $\mathcal{L}_\infty$  cases, lead to models with significantly different computational behavior. As a result, an exact (decomposition) approach, which perform very efficiently on one combination, often performs poorly with another combination of weights. It must be noted that although the polyhedral structure is not altered, the gradient of objective function becomes different for every combination of weights.

More precisely, although both problems possess structures that make primal decompositions such as Benders decomposition (Benders 2005) an obvious candidate, however, constraints (30) discourage Benders-like approaches for UDSFCHLP and  $\mathcal{L}_1$ . Because from among load balancing constraints (30)–(34) only constraints (30) (in addition to the subproblem described in Gelareh and Nickel (2011)) depend on the integer variables of the master and given the nature of such big-M constraints, the resulting Benders cuts become very weak—even after sharpening—and convergence tends to be very slow. For UDSFCHLP with  $\mathcal{L}_\infty$ , a combinatorial Benders Codato and Fischetti (2006) is a potential candidate rather than the classical Benders ( Benders (1962)) but even that did not report any good performance (with all sorts of tweaking, variable fixing, heuristic injections etc.).

For CDSHLP and  $\mathcal{L}_1$ , a very sophisticated Benders decomposition (equipped with variable elimination, specialized branching and various cut separations etc.) has not reported any particular efficiency. For CDSHLP and  $\mathcal{L}_\infty$ , we observed the same issues as with the case of UDSFCHLP with  $\mathcal{L}_\infty$ .

Likewise, some sophisticated dual approaches such as Lagrangian and column generation-like approaches have also been examined with very limited success being reported.

Nevertheless, the same properties that suggest a primal-type relaxation can be exploited to propose a heuristic approach.

## 4.1 Variable neighborhood methods

Variable Neighborhood Search (VNS) (Mladenović and Hansen 1997) is a flexible framework for building heuristics to solve combinatorial and continuous global optimization problems approximately. The idea is to systematically alternate exploring different neighborhood structures while seeking an optimal (or near-optimal) solution. The philosophy of VNS relies on the following facts: (i) a local optimum relative to one neighborhood structure is not necessarily the local optimum for another neighborhood structure; (ii) a global optimum is a local optimum with respect to all neighborhood structures; and (iii) for many problems, empirical evidence shows that all local optima are relatively close to each other.

In VNS, we alternate between improvement methods that are based on exploring different neighborhoods in a systematic way seeking better solutions and a shaking method in charge of injecting some noises (perturbation) before reaching predefined stopping condition(s).

In what concerns the *Improvement* methods, several systematic ways of exploring the neighborhood have been proposed in the literature: *sequential Variable Neighborhood Descent (seqVND)* [see e.g., (Hansen et al. 2010) for more details]; *Composite (Nested) VND* [see e.g., Todosijević et al. (2015) where full Nested VND was applied for the first time]; and *Mixed Nested VND* [see e.g., Ilić et al. (2010) for a recent application of Mixed Nested VND]. The so called General Variable Neighborhood Search (GVNS) (Hansen et al. 2010) is the one that incorporates advanced procedures in the improvement phase.

The *Shaking* procedure is applied to help escape from local optima where improvement procedures get stuck and the search process stagnates. Traditionally, two parameters are being used as stopping criteria: *maxIter*, the *maximum number of iterations* and *t<sub>max</sub>*, the *maximum allowed CPU time*. Additionally, we terminate the process if *maxNoImprove* has been passed and no improvement has been observed.

Interested readers are referred to (Hansen et al. 2010; Mladenović et al. 2016) for survey of successful applications of variants of VNS for solving different optimization problems.

Our approach that is elaborated in the following integrates an MIP solver as a part of the solution process whether for solving the purely continuous part of the problem or for recreating a partially ruined solution resulted by the operators. In such case, we use CPLEX as a component of the solution method. In this way, one may wish to consider this approach a matheuristic-like method.

## 4.2 Initial solution

Let for fixed values of variables  $h$  and  $y$ ,  $CDSHLP(\bar{h}, \bar{y})$  and  $UDSFCHLP(\bar{h}, \bar{y})$  be called  $\overline{CDSHLP}$  and  $\overline{UDSFCHLP}$ , respectively. For fixed set of hub nodes and hub edges,  $\overline{UDSFCHLP}$  has always a feasible solution for any  $1 \leq p \leq n$  and  $0 \leq q \leq \frac{p(p-1)}{2}$ . For  $\overline{CDSHLP}$ , a trivial solution is the one wherein all the edges are of hub type.

A *naïve* initial solution for any of the two problems can be constructed as in the following: (1) UDSFCHLP: choose any  $q$  edges among them (resulting in a connected hub-level sub-network). The  $q$  hub edges characterize the hub nodes and the hub-level structure of a solution and the remaining problem is solved using an LP solver, CPLEX in this case. Note that due to the availability of hub-level network structure, a preprocessing on the flow variables results in a significantly smaller LP, which can be solved very efficiently. (2) CDSHLP: set all the nodes as hub nodes with a complete graph. One then iteratively removes the hub edges carrying the least traffic one by one, and stop when the next removal will cause infeasibility. Again, the problem after fixing hub-level structure is solved as an LP by (with a similar preprocessing

applied to it) and in each iteration we warm start the optimal LP basis of the previous iteration, even further improving the efficiency in solution.

A more clever initial solution can be created as follows:

- UDSFCHLP: sort the O–D demands by their O–D flow volumes. Start from the busiest O–D, connect them with a direct hub edge and then choose the remaining edges from the top of the ordered list in such a way that the resulting hub level sub-network remains connected and the number of edges is .
- CDSHLP: solve the LP relaxation of the single objective median problem without load-balancing constraints. One then fixes the fractional (variable that takes a value between 0 and 1) hub nodes and hub edges (e.g. using a threshold) iteratively until reaches an integer feasible solution, which may not necessarily take into account load-balancing constraints. As a rule of thumb, a threshold of 0.85 and 0.15 for fixing to 1 and 0 is used. We give priority to fixing to 1 first.

In what follows, we opted to use the pair of clever algorithms as the quality of initial solutions generated by them are very often much better than the one from the alternative *naïve* method.

In all the aforementioned cases in this section and in what follows, unless stated otherwise, a reduced problem (wherein part of the variables are fixed by the local search operator) is being solved. The constraint sets for these problems are those of UDSFCHLP or CDSHLP and the objective function is that of compromise programming developed in Sect. 3, i.e., (61) for  $p = 1$  and (63) with two constraints (64) and (65) for  $p = \infty$ .

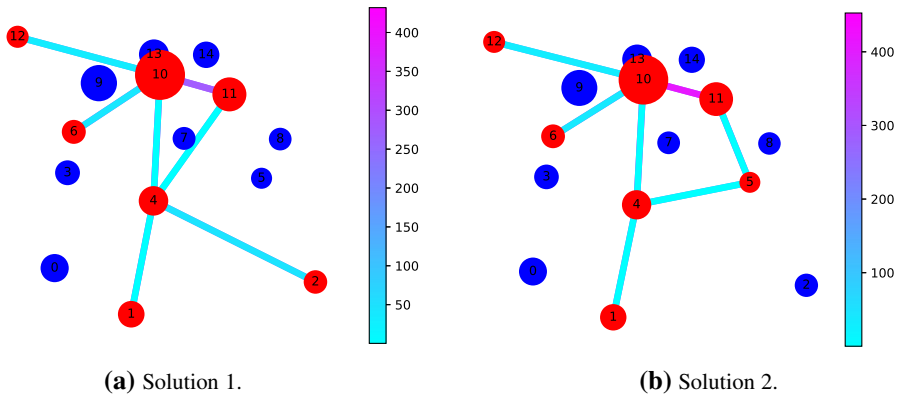
### 4.3 Neighborhood structures

In general two types of neighborhood structures are suggested. The first type deals with neighborhoods with respect to the location of hub nodes and the second type is concerned with the neighborhoods in relation with the edges.

#### 4.3.1 Node-based neighborhoods

The location neighborhoods used within our VNS are based on the following moves, which are defined with respect to a given solution  $s$ :

- DROP\_HUB( $i$ ): it removes a hub node  $i$  from the current solution  $s$ . If the gap between the highest ( $\mathcal{L}^h$ ) and the lowest ( $\mathcal{L}_l$ ) volume of flow transiting through hub nodes is significant ( $\frac{\mathcal{L}^h}{\mathcal{L}_l} > 2$ ) then the less busy node is turned into a spoke node. Otherwise, if the variance of loads is not significantly high, the candidate hub node is normally chosen randomly (with equal probability) from among the 50% hub nodes with the lowest utilization following. The logic behind this is that dropping a hub node will anyway result in the reduction of setup cost and given the bi-objective nature of the problem one can use this opportunity to move aligned with the load balancing objective function as well. Looking at Fig. 7b, one realizes that in this example, the nodes 5 and 6 (when compared to the node 10) are the candidates for such a move. In the case that the cardinality of hub edges is fixed to  $q$ , one can either introduce a busy node such as node 9 to be the new hub node or leave it to CPLEX to continue and complete the solution with a better one in the neighborhood of the partial solution (ruined one). However, using CPLEX, on one hand can be quite expensive and on the other hand, within a given time limit, may deliver high quality solutions that would be hard to find otherwise.



**Fig. 7** An example of a feasible solution. The node sizes are proportional to the total supply/demand/transshipment volumes and the (hub) edges colors correspond to the intensity of flow on the hub edges according to the color map. The red nodes represent the hub nodes and the blue ones represent the spoke nodes

- **ADD\_HUB**: if we give a higher weight to the median-type part of the objective function, it has been regularly observed through extensive computational experiments that a trade-off between cost and flow balance often (except in some extreme cases) encourages nodes with highest supply/demand to be among the hub nodes. Figure 7a also depicts another example of such nodes (#10 and #11) that appear as hub nodes. In practice, it is often the case that a hub is located at or very close to the location of major clients (one or more clients with maximum inward/outward flows). Therefore, one can either search for a hub node in the neighborhood of the busiest nodes or resort to a heuristic approach in which case one constructs a convex hull of the top 3 busiest (hub or non-hub) nodes and add one hub from among the ones closest to the perimeter of this set, if any, as a hub node. In the uncapacitated fixed cardinality version, when a new hub is added one must connect this new hub node to the other set and remove at least one edge in order to make sure that only  $q$  edges will remain in the network. The later can be selected from among those edges that have the least flow and whose removal does not split the hub-level network in two components. Alternatively, one can add local search constraints. Let  $\bar{Y}$  and  $\underline{Y}$  be the set of existing and potential (but currently non-hub) hub edges, respectively. A local search constraint  $\sum_{\{k,l\} \in \bar{Y}} (1 - y_{kl}) + \sum_{\{k,l\} \in \underline{Y}} y_{kl} \leq k$  for  $k = 2$  or  $3$  can be added and CPLEX can be called for a time limit to find a new good feasible solution. For the capacitated case, the addition of one hub node does not harm feasibility but may alter the objective function.
- **SWAP\_HUBS**: chooses a node to leave the set of hub nodes and a spoke to become a hub node. In Fig. 7a, the node #2, which is a hub node and connected to only one other hub node will become a spoke node and node #5 becomes a hub node. The primary edge encompassed from the new hub node connects the closest hub node (i.e. #11) from among the existing nodes. Once the hub node and the connection to the closest node have been identified, the other promising outgoing edges to the existing hub nodes (e.g. 5–4) may be identified by carrying out a heuristic local search within the edge-based neighborhoods or calling CPLEX by adding a local search constraint.

When using CPLEX to solve the reduced problem, we fix the lower/upper bounds on those variables where the corresponding elements is fixed by the heuristic.

### 4.3.2 Edge-based neighborhood

- DROP\_EDGE ( $i - j$ ): removes a hub edge  $i - j$  from the current solution  $s$ . The hub edge is normally chosen from among the hub edges with the lowest ratio  $\frac{\text{utilization}}{\text{fixedcost}}$  or  $\frac{\text{utilization}}{\text{capacity}}$  and with an equal probability.
- ADD\_EDGE: in the case of UDSFCHLP, this operator follows a call to DROP\_EDGE ( $i - j$ ). A prudent choice of a hub edge to be added is the one that can de-stress the hub edges with highest traffic by introducing a sort of redundancy. If possible, this can be done by adding an edge between the existing hub nodes that are receiving significant traffic. The second situation is when a hub must also be added. One then introduces a new hub node and adds an edge from that node to one of the closest hub nodes, which is chosen according to a uniform distribution. Again, our extensive computational experiments show that it is unlikely that this node is the closest node to the current set of hubs because the added value of introducing this edge (fixed cost for the new hub node and the hub edge) is rather negligible or even negative. Therefore, this new hub edge should have some meaningful minimum length. Our empirical conclusion arising from extensive computational experiments is that it must have a length at least equal to 10% of the distance between the two most remote nodes (diameter) of the network.
- SWAP\_EDGE: an under utilized hub edge with the lowest ratio  $\frac{\text{utilization}}{\text{fixed cost}}$  can be removed in favor of adding a direct connection between hub nodes  $i$  and  $k$  where  $i - j - k$  represent a sequence of two hub edges with the highest traffic on them. This introduces a redundancy and will de-stress  $i - j - k$ .

In all the aforementioned local search methods, in the initial iterations of the solution algorithm we try to avoid using CPLEX (mainly due to the computational overheads) as we still have several unseen feasible solutions in the search space that can improve the current solution. But as iterations continue and the success rate of heuristic raises, the use of CPLEX becomes much more viable as it can often find solutions that are not found by the other method.

### 4.3.3 Feasibility and objective value

If the hub-level network is connected and composed of only one component, we can then discuss the feasibility of  $\overline{CDSHLP}$  and  $\overline{UDSFCHLP}$ .  $\overline{CDSHLP}$  and  $\overline{UDSFCHLP}$ , which are then pure nonlinear programming formulations that can be solved by any solver for the second order cone programming such as CPLEX. In the case of  $\overline{UDSFCHLP}$ , the feasibility is always guaranteed as long as constraints (26) are satisfied, but in  $\overline{CDSHLP}$  the capacity constraint is the only source of infeasibility.

**Note 1** *It must be noted that as the flow subproblem is an expensive one to solve for every candidate solution, a hashing system can guarantee the efficiency and the success of this solution method.*

## 4.4 Adaptive variable neighborhood descent (AVND)

AVND works as following: Create and order a list of neighborhood structures and examine them in a linear order defined below. A state-transition matrix ( $\mathcal{T}$ ) is introduced where  $\mathcal{T}_{ij}$  represents the reward attributed when neighborhood  $j$  is examined after the neighborhood  $i$ .

After exploring the neighborhoods using a given order, a reward proportional to its performance is attributed to that operator and in the state-transition matrix influencing the chance of that operator for being solicited in the future iterations.

Let  $\ell_{max}$ -tuple  $\mathcal{N} = (\mathcal{N}_1, \dots, \mathcal{N}_{\ell_{max}})$  be an ordered set of operators defining the neighborhood structures. The sequential version works as following: we start with the first operator ( $l = 1$ ) and continue until all the operators are visited. Let  $l$  be the current operator and  $s$  be the current solution. We find the best solution in the neighborhood of  $s$  explored by the operator  $l$ . If this best neighbor improves the current solution, the current solution gets updated and we go back to the first operator  $l = 1$ , otherwise we move to the next operator. The process terminates when in a given iteration the current operator is  $l_{max}$  and no improvement is observed.

Our initial extensive computational experiments show that the most promising order of operators is (ADD\_HUB, ADD\_EDGE, DROP\_HUB, DROP\_EDGE, SWAP\_HUB, SWAP\_EDGE).

In the first round of our solution algorithm as we do not have any reward attributed to the operators, we run a sequential VND and calculate rewards. Once the rewards are calculated, we switch to an Adaptive VND (AVND) mode and thereupon the operators are selected based on the accumulated rewards attributed to them in the state-transition matrix  $\mathcal{T}$ . If rewards do not change significantly in the transition from one iteration to another, we call on the shaking phase.

## 4.5 Shaking procedure

The shaking function  $\text{Shake}(s, k)$  is called whenever the search is stagnated and no improvement is observed. It basically take two parameters—the solution  $s$  and an integer number  $k$ . There are two types of shaking: node shaking and edge shaking.

### 4.5.1 Node shaking

Let  $\overline{H}(s)$  be the set of hub nodes in  $s$  and  $\underline{H}(s)$  be the set of non-hub nodes in  $s$ .  $\text{Shake}$  randomly chooses  $k$  elements from  $\overline{H}$  and adds them to  $\underline{H}$  and at the same time chooses  $k$  nodes from  $\underline{H}$  and adds them to  $\overline{H}$ . The  $k$  new hub nodes are connected to their closest hub nodes using hub edges. We give higher weights to locate new hub nodes among the locations with less history of hub facilities among the visited solutions so far.

### 4.5.2 Edge shaking

The edge shaking follows the same strategy as the node shaking but we also need to make sure that the resulting hub level remains connected.

## 4.6 Steps of the proposed AVND

Algorithm 1 outlines the major steps of our AVND. The algorithm described in Sect. 4.4 serves as the local search part and the perturbation introduced in Sect. 4.5 acts as the diversification and shaking procedure.

AVND accepts two parameters, as shown in Algorithm 1:  $t_{max}$ , the total CPU time as time limit and  $k_{max}$  as the maximum number of iterations of the inner loop before resorting to the shaking procedure. Algorithm 1 shows that the starting solution is obtained using the

procedure detailed in Sect. 4.2 and it is subjected to some improvement using the sequential VND/AVND elaborated in Sect. 4.4.

---

### Algorithm 1 AVND

---

```

1: procedure AVND ( $k_{max}, t_{max}$ )
2:   generate initial solution using procedure described in Sect. 4.2
3:    $s \leftarrow \text{seqVND}(s)$  ▷ improvement using the seq. VND elaborated in Sect. 4.4
4:   repeat
5:      $k \leftarrow 1$ 
6:     while  $k \leq k_{max}$  do
7:        $s' \leftarrow \text{Shake}(s, k)$ 
8:        $s'' \leftarrow \text{seqVND}(s')$  ▷ improvement using the Adaptive VND elaborated in Section 4.4
9:        $k \leftarrow k + 1$ 
10:      if  $v(s'') < v(s)$  then
11:         $s \leftarrow s''; k \leftarrow 1$ 
12:      end if
13:    end while
14:  until  $\text{CpuTime}() > t_{max}$ 
15:   $t \leftarrow \text{CpuTime}()$ ;
16:  return  $s$ 
17: end procedure

```

---

## 5 Computational experiments

All the reported algorithms were coded and compiled under VC++ compiler in Windows 10 and run on a personal computer with an Intel Core i7-6700K CPU, 4.0 GHz and 32 GB of RAM. CPLEX 12.7.1 is used as an MIP solver.

In all our computational experiments we assumed that  $\alpha = 0.75$ . The parameter `IloCplex::BarQCPEpComp` has been set to  $10^{-9}$  and `IloCplex::MIPEmphasis` has been set to `CPX_MIPEMPHASIS_BALANCED`. The default value for `IloCplex::Param::MIP::Strategy::MIQCPStrat` is maintained. The custom termination criteria for CPLEX (UserAbort) is the following: we terminate CPLEX if a maximum of 10h (36,000s) of CPU time is passed and an incumbent solution has been identified. If within this 10h no solution is found then it continues until either the first feasible solution is found or CPLEX crashes due to an exception. While this 36,000s is imposed, CPLEX only checks the CPU time upon termination of the current node LP resolve. This may result in a CPU time slightly larger than 36,000s in our tables. For the heuristic approach, we also use the same time limit, although the process terminates in much less time.

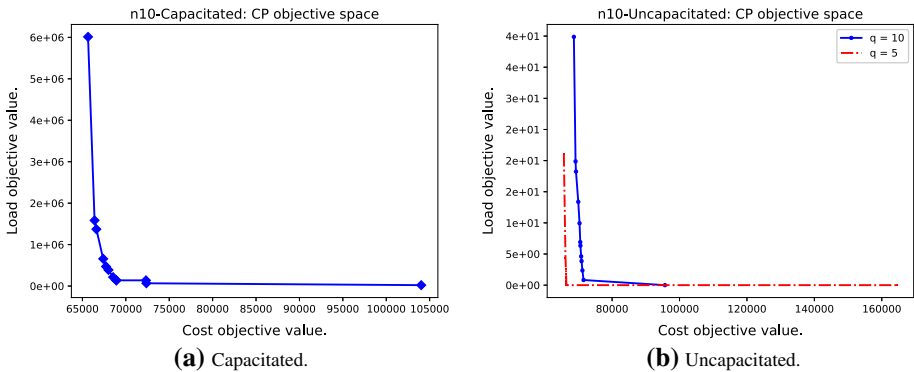
The capacity of hub edges is chosen uniformly by  $U[0.1, 0.15] \times \sum_{i,j} w_{ij}$ . For the heuristic,  $k_{max}$  is set to  $n$  and  $t_{max}$  is set to 7,200s. *maxNoImprove* is also set to 1000. For the uncapacitated case, the constraint (58) has been added to the model. The first problem being solved is the cost optimization problem and its solution is being used to warm start the other problem with  $w_1 = 0.1$ . For every other  $w$ , the CPLEX is warm-started using the solution for the previous  $w_i$ . Note that  $w_i \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ . A hashing system has been used to retrieve the objective solution should the problem has been solved previously.

### 5.1 Testbed

To carry out our computational experiments, we use the instances from a real practice data set of a telecommunication operator. The operator is a major one—with an extended network

**Table 2** The  $n$  values and their corresponding  $q$  values

# Nodes	# Edges
$n = 10$	$q = 5, 10$
$n = 15$	$q = 5, 10, 15, 20$
$n = 20$	$q = 5, 10, 15, 20, 25$
$n = 25$	$q = 5, 10, 15, 20, 25, 30$
$n = 30$	$q = 5, 10, 15, 20, 25, 30$



**Fig. 8** The CP frontiers corresponding to the CDSHLP and DSHLP instances with  $n = 10$

from central Europe to the Suez Canal and widespread across Africa—that also owns a major part of the national infrastructure. Therefore, almost any operator in this country is leasing capacity on the network maintained by this major operator. The clients are of two categories: other equally popular operators that have acquired their own dedicated bandwidths on long-term lease contracts and the low-cost ones that are dynamically leasing residual capacities from the unused segments of the operational networks. Therefore, the infrastructure owner is not in charge of congestion pricing but a longer term strategic network design capable of assuring a balanced load to cope with all such demands given some historical information, the current trend of technology/demand and the forecast and prediction for the future.

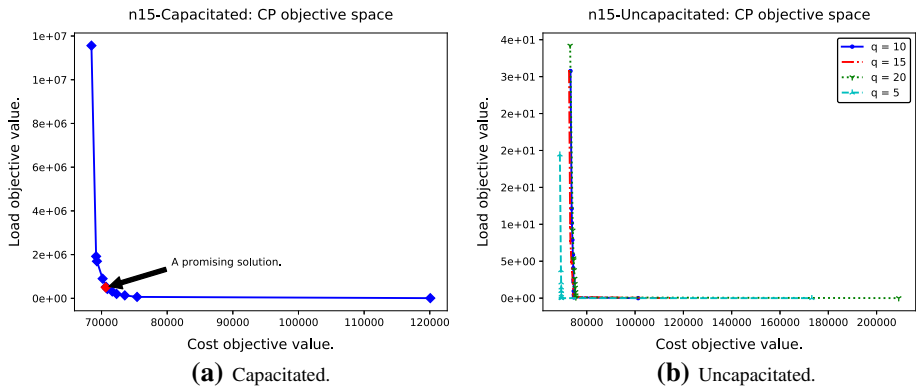
Our main instance is a network composed of 30 nodes scattered across the country within the national-level network. These 30 nodes include the major hubs (hub nodes) and secondary hubs (spoke nodes) which are the subject of this network design problem. To create smaller instances, we use a kind of aggregation [akin to the one proposed by Ernst and Krishnamoorthy (1996)] to establish grids on the map and introduce new virtual nodes as the centers of gravity of cells in the grid with a demand equal to that of all the nodes within the grid cell.

No more than half of these nodes are expected to be used as hub nodes and the number of edges (technological connections) is always limited to at most 30 edges. Therefore, in practice the optimal number of hub is between 5 and  $\frac{n}{2}$ . For every  $n$ , the corresponding  $q$  values are indicated in Table 2.

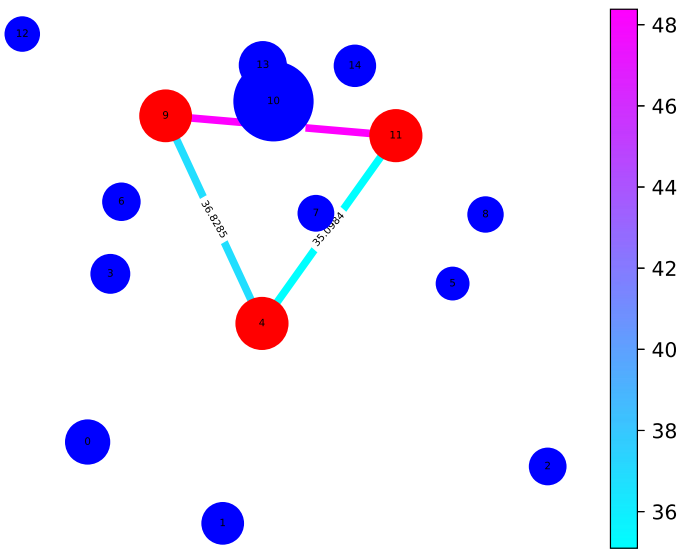
### 5.2 Analysis

In Fig. 8a, b, the CP frontiers corresponding to Fig. 6 are depicted. In Fig. 8b, we plot all the values for  $q$  in Table 2.





**Fig. 9** The CP frontiers corresponding to the CDSHLP and DSHLP instances with  $n = 15$



**Fig. 10** The decision space representation of the promising member of frontier for  $n15$  in Fig. 9a

The curve corresponding to the capacitated version defines a smoother representation of the frontier. In the case of uncapacitated, clearly  $q = 5$  absolutely dominates  $q = 10$  in terms of cost and flow balance. It can be concluded that adding more than 5 hub edges will not necessarily result in a better trade-off between cost and load balancing for this instance.

Similar behavior can be observed in Fig. 9a, b (respectively, in Fig. 11a, b). Again the capacitated version defines a smoother curve. In Fig. 9b (Fig. 11b), the image combines plots for all the values of  $q$  corresponding to this instance in Table 2. As it can be observed, again  $q = 5$  dominates all the other values of  $q$  for this instance in Table 2.

Figure 10 depicts a decision space representation of a solution on the frontier in Fig. 9a.

For  $n = 20$ , a similar observation is depicted in Fig. 11.

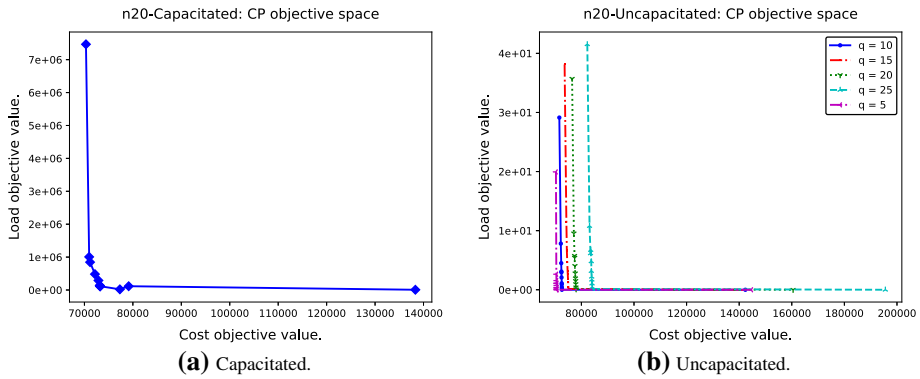


Fig. 11 The CP frontiers corresponding to the CDSHLP and DSHLP instances with  $n = 20$

### 5.3 CPLEX versus heuristic

Tables 3, 4, 5 and 6 report the computational results for instance of 10, 15 and 20 nodes in CDSHLP and DSHLP, respectively. Every table is divided into two blocks: CPLEX results and Heuristic results. In the first block, the first column indicates the instance name in the format “nX-c-wY-ZZZ” (“nX-wY-ZZZ” for the uncapacitated case ) where ‘X’ represent number of nodes, ‘Y’ stands for the coefficient of w in the CP model and ‘ZZZ’ to identify whether it was  $\mathcal{L}_\infty$ ,  $\mathcal{L}_1$ , the cost or load balancing objective functions. The second column (‘TimeIP’) reports the CPU time (in seconds) for solving that case. The next column (‘Nnodes’) indicates the number of branch-and-bound nodes processed followed by the column reporting the gap (‘Gap’) reached during the time. The column (‘IPStatus’) stands for the status of CPLEX upon termination followed by the reported objective value in the next column (‘IPobj’). The columns (‘Z<sub>1</sub>’) and (‘Z<sub>2</sub>’) report the split objective values for cost and load balancing separately. The last two columns (‘#Hubs’) and (‘#Edges’) indicate the number of hub nodes and hub edges reported by the corresponding solution delivered by CPLEX. The tables are composed of blocks for every instance and different objective function corresponding to (cost, load balancing and different values of  $w_i$ ). In the second block, the first column (‘TimeH’) represents the time (in second) within which the heuristic methods has terminated. The second column (‘TimeHB’) represent the time (in seconds) at which the best solution of heuristic has been identified and the last column (‘GapH (%)’) represents the gap between the solution of heuristic relative to the best reported bound of the exact method.

For every instance configuration ( $n$  with given parameters) there are four rows appended to report a summary of results for the given set of instances. The first row reports the average CPU time spent in CPLEX solving reduce problem for the given  $n$ . In the second to fourth rows depending on whether it is capacitated or uncapacitated case, we report different values. For UDSFCHLP, the second to the fourth rows report the ratio of volumes on the less loaded hub edge and the busiest one and the variance of loads on hub nodes (separated by a slash) for the cases of cost-only (VarH (C)), load balancing (VarH (L)) and the average over all other cases (i.e. VarH ( $\mathcal{L}_1$ ,  $\mathcal{L}_\infty$ )). For CDSHLP, three consecutive lines report the minimum, maximum and average utilization (separated by slash) as well as variance of loads for the cases of cost-only, load balancing and the average over all other cases.

As one observes, the ideal load balancing objective function tends to install the minimum required number of hub edges, i.e. one. A similar behavior can be observed in  $\mathcal{L}_1$  when

the weight corresponding to the load balancing objective function increases and that of cost decreases.

In the CP objective function as in (59),  $w_i$  refers to the weight of cost objective function and  $1 - w_i$  to the load balancing objective. In Table 3, for every  $n$ , the higher coefficient  $w_i$  suggest opening more hub nodes and hub edges and the smaller values tends to open less number of hub nodes. The computational times are relatively moderate and the number of processed branch-and-bound nodes are very few indicating that processing nodes in the branch-and-bound tree is the most time consuming task. All instances, except one, were solved to the optimality. Often the solution found in the early iterations were optimal but it has taken long for CPLEX to prove the optimality. Therefore, given that our heuristic also did not find any better solution for this particular instance, we expect this also to be the optimal even though the proof is not at hand.

It must be noted that in (59), except for case of ‘idealCost’, no more than 5 hub nodes has been opened in any solution.

In Tables 4, 5 and 6, we deal with the uncapacitated version which takes into account a fixed cardinality of hub edges and an upper bound on the number of possible hub nodes (at most half the number of nodes). As one observes the computational times for  $\mathcal{L}_1$  instances are significantly less than those for  $\mathcal{L}_\infty$ . In general, computational time increases exponentially as the instance size increases. The smaller number of processed branch-and-bound nodes confirms that the problem being solved at every node is indeed a challenging one resulting in less number of nodes being processed within the time limit. Sometimes in this column values larger than our time limit are reported this is because when the time limit has been reached, CPLEX was in the middle of solving the node sub-problem and termination process for that node took very long time.

As before we solve the problems in this order: first ‘idealCost’ was solved and then the obtained optimal solution was used to warm start CPLEX for solving ‘idealLdBIncng’. After that, this warm start style has been repeated for all other problems of  $\mathcal{L}_1$  and  $\mathcal{L}_\infty$ , respectively, in the given order. Our computational experiments confirm that the use of the previous solution to help warm-start CPLEX indeed reduced the required computational efforts very significantly. Often such solutions were optimal or very close to optimal but it took very long time for CPLEX to prove that. In this table, all instances, except six, were solved to optimality within the time limit. Allowing longer run time showed that indeed some of the six cases reported solutions even though showing high gap in this table, they are indeed either optimal or within less than 5% of the optimality when optimality was proven beyond the time limit.

For the uncapacitated problem, UDSFCHLP, for any feasible network structure, there is always an optimal solution with the objective value of 0. When comparing the number of hub nodes and hub edges among different instances of uncapacitated version, one observes that while for the cases of  $n = 10, 15$  for the given  $q$ , the maximum number of hub nodes (i.e.  $\frac{n}{2}$ ) is being opened, for  $n=20$ ,  $q = 10, 15$ , always 8 and 9, hub nodes (i.e. less than  $\frac{n}{2}$ ), respectively, were opened for CP approach and the given values of  $q = 10, 15$  (the two other exceptions for every  $q \in \{10, 15\}$  are the single objective cases corresponding to ‘idealCost’ and ‘idealLdBIncng’). However, again starting from  $q = 20$  the number of hub nodes being opened increases as  $q$  increases.

**Table 3** Numerical experiments for the CDSHLP instances ( $n = 10, 15$  and  $20$ )

Instance	CPLEX										Heuristic	
	TimeIP	Nnodes	Gap (%)	IPStatus	IPobj	$Z_1$	$Z_2$	#Hubs	#Edges	TimeH	TimeHB	GapH(%)
n10-c-idealL4Blncg	9.62	17	0.00	OptimalTol	21,734.09	104,027.95	21,734.09	2	1	0.19	0.01	0.00
n10-c-idealCost	1.01	0	0.00	Optimal	65,647.36	65,647.36	6012,296.25	5	6	0.21	0.01	0.00
n10-c-w9- $\mathcal{L}_1$	7.38	24	0.00	Optimal	43.72	66,399.56	1583,893.95	5	6	0.21	0.01	0.00
n10-c-w9- $\mathcal{L}_\infty$	19.45	28	0.00	Optimal	225.26	66,607.99	1371,174.47	5	6	0.36	0.01	0.00
n10-c-w7- $\mathcal{L}_1$	11.12	41	0.00	Optimal	60.20	67,714.87	470,895.28	4	4	0.13	0.03	0.00
n10-c-w7- $\mathcal{L}_\infty$	6.86	43	0.00	Optimal	318.01	67,390.99	656,747.34	4	4	0.18	0.01	0.00
n10-c-w5- $\mathcal{L}_1$	3.15	14	0.00	Optimal	52.17	68,908.41	137,808.76	3	3	0.21	0.01	0.00
n10-c-w5- $\mathcal{L}_\infty$	11.11	57	0.00	Optimal	306.33	67,998.82	388,756.84	4	5	0.53	0.01	0.00
n10-c-w3- $\mathcal{L}_1$	3.98	14	0.00	Optimal	39.05	68,908.41	137,808.76	3	3	0.23	0.03	0.00
n10-c-w3- $\mathcal{L}_\infty$	10.12	58	0.00	Optimal	225.74	68,535.25	214,917.09	3	2	1.21	0.00	0.00
n10-c-w1- $\mathcal{L}_1$	4.27	29	0.00	OptimalTol	24.30	72,366.02	66,978.62	2	1	0.44	0.07	0.00
n10-c-w1- $\mathcal{L}_\infty$	10.55	73	0.00	Optimal	173.81	72,318.43	137,427.53	2	1	0.96	0.21	0.00

Table 3 continued

Instance	CPLEX				Heuristic							
	TimeIP	Nnodes	Gap (%)	IPStatus	IPobj	Z <sub>1</sub>	Z <sub>2</sub>	#Hubs	#Edges	TimeH	TimeHB	GapH(%)
n15-c-ideal	379.02	24	0.00	Optimal	8608.43	120078.41	8608.43	2	1	4.32	2.81	0.001 s
n15-c-ideal	15.90	0	0.00	Optimal	68460.95	68460.95	11560472.95	6	5	1.44	1.24	(0.326,0.652,0.893)33.208
n15-c-w9- $\mathcal{L}_1$	62.79	102	0.00	Optimal	28.97	69173.54	1920014.21	5	4	4.68	3.88	(0.639,0.667,0.697)1.854
n15-c-w9- $\mathcal{L}_\infty$	158.20	65	0.00	Optimal	146.00	69298.27	1695118.21	5	4	3.34	3.17	Avg. Min/Avg/Max/VarH ( $\mathcal{L}_1, \mathcal{L}_\infty$ ): (0.511,0.646,0.723)7.537
n15-c-w7- $\mathcal{L}_1$	51.42	108	0.00	Optimal	43.53	70679.01	526410.33	4	3	1.80	0.85	0.00
n15-c-w7- $\mathcal{L}_\infty$	175.56	200	0.00	OptimalTol	231.25	70166.20	899081.00	4	3	0.96	0.12	0.00
n15-c-w5- $\mathcal{L}_1$	29.48	103	0.00	OptimalTol	41.69	70883.92	429549.26	4	3	4.00	0.56	0.00
n15-c-w5- $\mathcal{L}_\infty$	81.90	84	0.00	Optimal	217.66	70707.93	511476.20	4	3	0.47	0.06	0.00
n15-c-w3- $\mathcal{L}_1$	71.15	193	0.00	OptimalTol	33.92	72294.87	200660.52	4	3	4.32	1.90	0.00
n15-c-w3- $\mathcal{L}_\infty$	353.94	254	0.00	Optimal	184.78	71640.04	313545.52	4	3	2.40	1.56	0.00
n15-c-w1- $\mathcal{L}_1$	83.77	107	0.00	OptimalTol	17.84	75412.66	64690.82	3	2	7.68	5.45	0.00
n15-c-w1- $\mathcal{L}_\infty$	331.02	204	0.00	Optimal	98.41	73540.21	134925.14	3	3	5.85	3.57	0.00

Table 3 continued

Instance	CPLEX					Heuristic						
	TimeIP	Nnodes	Gap (%)	IPStatus	IPobj	Z <sub>1</sub>	Z <sub>2</sub>	#Hubs	#Edges	TimeH	TimeHB	GapH (%)
n20-c-ideal	11157.07	34	0.00	Optimal	7590.77	138318.43	7590.77	2	1	266.21	252.70	0.023 s
n20-c-ideal	114.58	35	0.00	Optimal	70314.10	7467491.36	7	6	474.20	222.78	0.00	(0.361,0.592,0.974)101.413
n20-c-w9-L <sub>1</sub>	404.79	96	0.00	OptimalTol	22.15	70976.12	1006061.87	5	4	294.01	161.70	0.00
n20-c-w9-L <sub>∞</sub>	28059.82	379	0.00	Optimal	112.42	71163.54	846229.98	5	4	393.07	255.45	0.00
n20-c-w7-L <sub>1</sub>	459.44	312	0.00	OptimalTol	34.37	73232.75	115184.59	4	3	207.87	91.08	0.00
n20-c-w7-L <sub>∞</sub>	25784.56	801	0.00	Optimal	190.97	72169.43	482458.56	5	4	630.33	245.70	0.00
n20-c-w5-L <sub>1</sub>	178.00	71	0.00	OptimalTol	28.67	73232.75	115184.55	4	3	364.90	116.48	0.00
n20-c-w5-L <sub>∞</sub>	33921.09	366	0.00	Optimal	189.54	72891.98	290375.10	4	3	179.05	107.40	0.00
n20-c-w3-L <sub>1</sub>	163.27	179	0.00	Optimal	22.97	73232.75	115184.55	4	3	408.13	191.76	0.00
n20-c-w3-L <sub>∞</sub>	13424.56	124	0.00	Optimal	125.52	73161.47	141357.55	4	3	189.11	177.66	0.00
n20-c-w1-L <sub>1</sub>	2752.39	253	0.00	OptimalTol	11.88	77316.06	20733.48	2	1	202.68	105.04	0.00
n20-c-w1-L <sub>∞</sub>	8918.50	3	96.00	AbortUser	129.80	79133.93	115181.97	4	3	132.99	33.00	14.43
												Avg. CPLEX time in H: 0.752 s
												Min/Avg/Max/VarH (C): (0.411,0.505,0.967)214.370
												Min/Avg/Max/VarH (L): (0.437,0.486,0.527)11.380
												Avg. Min/Avg/Max/VarH (L <sub>1</sub> , L <sub>∞</sub> ): (0.340,0.515,0.780)45.730

**Table 4** Numerical experiments for the UDSFCHLP instances

Instance	CPLEX										Heuristic	
	TimeIP	Nnodes	Gap (%)	IPStatus	IPobj	Z <sub>1</sub>	Z <sub>2</sub>	#Hubs	#Edges	TimeH	TimeHB	GapH(%)
n10-q5-idealLdBhncg	0.23	0	0.00	OptimalTol	0.00	164516.96	0.00	5	5	0.96	0.18	0.00
n10-q5-idealCost	2.62	0	0.00	Optimal	65687.88	65687.88	441.20	5	5	0.34	0.04	0.00
n10-q5-w9-L <sub>1</sub>	34.62	572	0.00	OptimalTol	6.69	66380.38	1.69	5	5	0.62	0.51	0.00
n10-q5-w9-L <sub>∞</sub>	25.48	76	0.00	Optimal	44.59	66177.58	19.68	5	5	0.08	0.05	0.00
n10-q5-w7-L <sub>1</sub>	13.28	62	0.00	OptimalTol	5.42	66442.55	0.11	5	5	0.58	0.13	0.00
n10-q5-w7-L <sub>∞</sub>	26.18	37	0.00	Optimal	43.42	66300.96	6.39	5	5	0.78	0.41	0.00
n10-q5-w5-L <sub>1</sub>	11.86	53	0.00	OptimalTol	3.90	66454.99	0.02	5	5	0.32	0.16	0.00
n10-q5-w5-L <sub>∞</sub>	27.08	39	0.00	Optimal	33.65	66352.93	2.97	5	5	0.39	0.38	0.00
n10-q5-w3-L <sub>1</sub>	12.59	43	0.00	Optimal	2.35	66460.32	0.00	5	5	0.68	0.51	0.00
n10-q5-w3-L <sub>∞</sub>	22.75	99	0.00	Optimal	21.30	66389.42	1.34	5	5	0.31	0.27	0.00
n10-q5-w1-L <sub>1</sub>	10.76	39	0.00	OptimalTol	0.79	66463.28	0.00	5	5	0.70	0.43	0.00
n10-q5-w1-L <sub>∞</sub>	20.03	27	0.00	OptimalTol	7.46	66425.22	0.37	5	5	0.95	0.63	0.00

Table 4 continued

Instance	CPLEX										Heuristic		
	TimeIP	Nnodes	Gap (%)	IPStatus	IPobj	Z <sub>1</sub>	Z <sub>2</sub>	#Hubs	#Edges	TimeH	TimeHB	TimeH	GapH (%)
n10-q10-idealLdBIncng	0.34	0	0.00	OptimalTol	0.00	95676.18	0.00	5	10	9.40	9.31	Avg. CPLEX time in H:	0.012 s
n10-q10-idealCost	4.71	0	0.00	Optimal	68661.61	68661.61	1589.90	5	10	9.30	1.30	$\frac{Min}{Max}$ /VarH (C):	0.301/75.16
n10-q10-w9-L <sub>1</sub>	13.96	112	0.00	Optimal	41.41	69158.47	395.20	5	10	23.50	23.50	$\frac{Min}{Max}$ /VarH (L):	0.976/5.22
n10-q10-w9-L <sub>∞</sub>	29.94	91	0.00	Optimal	209.33	69289.93	332.81	5	10	38.40	8.06	Avg. $\frac{Min}{Max}$ /VarH (L <sub>1</sub> , L <sub>∞</sub> ):	0.731/21.034
n10-q10-w7-L <sub>1</sub>	38.05	447	0.00	OptimalTol	57.95	70605.85	40.14	5	10	15.30	8.57		0.00
n10-q10-w7-L <sub>∞</sub>	42.46	243	0.00	OptimalTol	337.92	69965.70	179.08	5	10	40.00	34.00		0.00
n10-q10-w5-L <sub>1</sub>	34.60	652	0.00	OptimalTol	46.64	70817.59	21.42	5	10	43.75	7.44		0.00
n10-q10-w5-L <sub>∞</sub>	50.22	593	0.00	OptimalTol	311.00	70341.92	98.89	5	10	50.00	6.00		0.00
n10-q10-w3-L <sub>1</sub>	35.57	540	0.00	Optimal	30.61	71196.56	5.58	5	10	39.75	31.80		0.00
n10-q10-w3-L <sub>∞</sub>	51.37	559	0.00	Optimal	210.34	70555.75	47.77	5	10	45.00	30.60		0.00
n10-q10-w1-L <sub>1</sub>	43.48	981	0.00	Optimal	11.03	71537.83	0.68	5	10	37.20	27.90		0.00
n10-q10-w1-L <sub>∞</sub>	49.97	464	0.00	Optimal	84.34	70940.00	14.90	5	10	15.00	7.20	Avg. CPLEX time in H:	0.244 s
												$\frac{Min}{Max}$ /VarH (C):	0.414/105.01
												$\frac{Min}{Max}$ /VarH (L):	0.989/20.110
												Avg. $\frac{Min}{Max}$ /VarH (L <sub>1</sub> , L <sub>∞</sub> ):	0.844/44.743



**Table 5** Numerical experiments for the UDSFCHLP instances

Instance	CPLEX										Heuristic	
	TimeIP	Nnodes	Gap (%)	IPStatus	IPobj	Z <sub>1</sub>	Z <sub>2</sub>	#Hubs	#Edges	TimeH	TimeHB	GapH(%)
n15-q5-idealLdBhncg	2.54	0	0.00	OptimalTol	0.00	156427.43	0.00	6	5	38.34	16.49	0.00
n15-q5-idealCost	67.58	47	0.00	Optimal	68460.95	68460.95	332.29	6	5	19.46	7.59	0.00
n15-q5-w9-L <sub>1</sub>	577.78	430	0.00	OptimalTol	7.98	69233.23	0.25	6	5	186.44	89.49	0.00
n15-q5-w9-L <sub>∞</sub>	1730.32	247	0.00	Optimal	55.59	69004.26	18.47	6	5	134.99	67.50	0.00
n15-q5-w7-L <sub>1</sub>	268.56	352	0.00	OptimalTol	6.25	69243.98	0.02	6	5	310.13	307.03	0.00
n15-q5-w7-L <sub>∞</sub>	1711.27	1009	0.00	Optimal	56.78	69174.47	6.29	6	5	89.83	76.35	0.00
n15-q5-w5-L <sub>1</sub>	278.38	315	0.00	Optimal	4.47	69246.13	0.00	6	5	214.95	51.59	0.00
n15-q5-w5-L <sub>∞</sub>	1677.28	297	0.00	Optimal	41.95	69198.96	2.79	6	5	149.20	68.63	0.00
n15-q5-w3-L <sub>1</sub>	468.25	455	0.00	Optimal	2.68	69247.05	0.00	6	5	370.15	344.24	0.00
n15-q5-w3-L <sub>∞</sub>	1577.62	488	0.00	Optimal	25.73	69215.45	1.22	6	5	85.23	10.23	0.00
n15-q5-w1-L <sub>1</sub>	251.15	360	0.00	OptimalTol	0.89	69247.57	0.00	6	5	193.68	147.20	0.00
n15-q5-w1-L <sub>∞</sub>	1843.21	390	0.00	OptimalTol	8.76	69231.13	0.32	6	5	84.43	58.26	0.00

Table 5 continued

Instance	CPLEX						Heuristic					
	TimeIP	Nnodes	Gap (%)	IPStatus	IPobj	Z <sub>1</sub>	Z <sub>2</sub>	#Hubs	#Edges	TimeH	TimeHB	GapH (%)
n15-q10-idealLdBIncng	2.48	0	0.00	OptimalTol	0.00	197782.63	0.00	7	10	46.72	15.88	0.00
n15-q10-idealCost	52.28	100	0.00	Optimal	69723.79	69723.79	2369.79	7	10	192.54	86.64	0.00
n15-q10-w9-L <sub>1</sub>	1098.70	1961	0.00	OptimalTol	7.06	70580.54	24.57	7	10	56.63	28.32	0.00
n15-q10-w9-L <sub>∞</sub>	2738.46	1797	0.00	OptimalTol	41.60	70315.68	98.58	7	10	87.95	29.03	0.00
n15-q10-w7-L <sub>1</sub>	958.16	2485	0.00	Optimal	6.13	70801.89	1.88	7	10	85.36	28.17	0.00
n15-q10-w7-L <sub>∞</sub>	2175.90	1548	0.00	OptimalTol	43.84	70525.87	34.63	7	10	6.09	4.45	0.00
n15-q10-w5-L <sub>1</sub>	1018.41	2549	0.00	Optimal	4.48	70851.65	0.36	7	10	156.30	29.70	0.00
n15-q10-w5-L <sub>∞</sub>	2506.34	2981	0.00	Optimal	35.47	70632.23	16.81	7	10	274.41	194.83	0.00
n15-q10-w3-L <sub>1</sub>	965.10	2603	0.00	Optimal	2.71	70873.65	0.07	7	10	105.55	45.39	0.00
n15-q10-w3-L <sub>∞</sub>	2621.88	3214	0.00	Optimal	23.15	70712.12	7.84	7	10	164.36	105.19	0.00
n15-q10-w1-L <sub>1</sub>	889.52	2297	0.00	Optimal	0.91	70885.87	0.00	7	10	148.84	86.33	0.00
n15-q10-w1-L <sub>∞</sub>	3310.29	2690	0.00	OptimalTol	8.36	70794.67	2.20	7	10	398.03	286.58	0.00

Table 5 continued

Instance	CPLEX										Heuristic		
	TimeIP	Nnodes	Gap (%)	IPStatus	IPobj	Z <sub>1</sub>	Z <sub>2</sub>	#Hubs	#Edges	TimeH	TimeHB	TimeH	GapH (%)
n15-q15-idealLdBIncng	3.00	0	0.00	OptimalTol	0.00	114593.03	0.00	7	15	206.17	134.01	Avg. CPLEX time in H:	0.820 s
n15-q15-idealCost	64.72	44	0.00	Optimal	72672.30	72672.30	946.30	7	15	143.49	53.09	$\frac{Min}{Max}/\text{VarH (C):}$	0.316/1023.440
n15-q15-w9-L <sub>1</sub>	303.25	714	0.00	Optimal	18.54	73240.01	60.06	7	15	185.73	183.87	$\frac{Min}{Max}/\text{VarH (L):}$	0.965/178.002
n15-q15-w9-L <sub>∞</sub>	1687.07	737	0.00	Optimal	97.38	73125.89	92.15	7	15	300.65	48.10	Avg. $\frac{Min}{Max}/\text{VarH (L}_1, \text{L}_\infty\text{):}$	0.893/287.230
n15-q15-w7-L <sub>1</sub>	569.17	1182	0.00	Optimal	24.36	73667.15	24.42	7	15	263.48	158.09		0.00
n15-q15-w7-L <sub>∞</sub>	1054.69	507	0.00	Optimal	125.30	73422.67	39.52	7	15	232.99	198.04		0.00
n15-q15-w5-L <sub>1</sub>	693.44	1736	0.00	Optimal	20.77	74158.04	5.77	7	15	288.79	176.16		0.00
n15-q15-w5-L <sub>∞</sub>	1394.94	650	0.00	Optimal	121.81	73693.54	23.05	7	15	228.83	43.48		0.00
n15-q15-w3-L <sub>1</sub>	751.07	1436	0.00	OptimalTol	13.17	74477.59	0.34	7	15	197.87	146.43		0.00
n15-q15-w3-L <sub>∞</sub>	1840.11	1536	0.00	Optimal	90.68	73939.45	12.26	7	15	213.75	117.56		0.00
n15-q15-w1-L <sub>1</sub>	924.87	1702	0.00	OptimalTol	4.45	74529.81	0.02	7	15	332.72	239.56		0.00
n15-q15-w1-L <sub>∞</sub>	1620.32	1606	0.00	OptimalTol	37.59	74248.20	3.95	7	15	105.88	49.76	Avg. CPLEX time in H:	0.978 s
												$\frac{Min}{Max}/\text{VarH (C):}$	0.266/1881.011
												$\frac{Min}{Max}/\text{VarH (L):}$	0.964/257.116
												Avg. $\frac{Min}{Max}/\text{VarH (L}_1, \text{L}_\infty\text{):}$	0.851/355.056

**Table 6** Numerical experiments for the UDSFCHLP instances

Instance	CPLEX											Heuristic	
	TimeIP	Nnodes	Gap (%)	IPStatus	IPobj	Z <sub>1</sub>	Z <sub>2</sub>	#Hubs	#Edges	TimeH	TimeHB	GapH(%)	
n20-q5-idealLdBhncg	15.66	0	0.00	OptimalTol	0.00	144583.82	0.00	6	5	2840.79	340.89	0.00	
n20-q5-idealCost	1078.42	1042	0.00	OptimalTol	70325.77	70325.77	397.47	6	5	2520.20	1436.51	0.00	
n20-q5-w9-L <sub>1</sub>	1784.65	257	0.00	Optimal	2.89	70538.24	1.26	6	5	1035.66	445.33	0.00	
n20-q5-w9-L <sub>∞</sub>	19460.38	253	0.00	Optimal	17.42	70469.48	6.92	6	5	3125.79	2969.50	0.00	
n20-q5-w7-L <sub>1</sub>	2162.35	629	0.00	Optimal	2.46	70578.13	0.11	6	5	4550.91	4141.33	0.00	
n20-q5-w7-L <sub>∞</sub>	29762.70	211	0.00	Optimal	18.15	70518.26	2.40	6	5	3960.40	1465.35	0.00	
n20-q5-w5-L <sub>1</sub>	1859.19	418	0.00	Optimal	1.79	70587.81	0.02	6	5	1470.41	632.28	0.00	
n20-q5-w5-L <sub>∞</sub>	30911.41	272	0.00	Optimal	14.46	70540.54	1.15	6	5	1480.91	192.52	0.00	
n20-q5-w3-L <sub>1</sub>	1704.14	235	0.00	Optimal	1.08	70591.93	0.00	6	5	1210.99	702.37	0.00	
n20-q5-w3-L <sub>∞</sub>	26233.47	578	0.00	OptimalTol	9.35	70557.15	0.53	6	5	1030.70	463.81	0.00	
n20-q5-w1-L <sub>1</sub>	1465.39	201	0.00	Optimal	0.36	70594.32	0.00	6	5	3920.50	1528.99	0.00	
n20-q5-w1-L <sub>∞</sub>	38385.57	954	0.00	Optimal	3.36	70575.02	0.15	6	5	1395.96	851.53	0.00	

Table 6 continued

Instance	CPLEX							Heuristic				
	TimeIP	Nnodes	Gap (%)	IPStatus	IPobj	Z <sub>1</sub>	Z <sub>2</sub>	#Hubs	#Edges	TimeH	TimeHB	CapH(%)
n20-q10-idealLdBIncng	16.02	0	0.00	OptimalTol	0.00	142345.81	0.00	9	10	2820.05	2171.44	0.00
n20-q10-idealCost	1570.51	339	0.00	OptimalTol	71609.89	71609.89	848.82	9	10	1580.96	1533.53	0.00
n20-q10-w9-L <sub>1</sub>	26759.27	3429	0.00	OptimalTol	11.14	72404.45	8.72	8	10	2950.25	1829.15	0.00
n20-q10-w9-L <sub>∞</sub>	82790.23	2476	0.00	OptimalTol	71.85	72174.60	60.99	8	10	2525.50	2222.44	0.00
n20-q10-w7-L <sub>1</sub>	16154.11	2962	0.00	OptimalTol	9.32	72523.29	0.80	8	10	3760.99	2482.25	0.00
n20-q10-w7-L <sub>∞</sub>	100490.85	1389	0.00	Optimal	71.54	72332.84	20.24	8	10	1740.40	539.52	0.00
n20-q10-w5-L <sub>1</sub>	22702.47	2719	0.00	Optimal	6.80	72556.43	0.19	8	10	1650.84	660.33	0.00
n20-q10-w5-L <sub>∞</sub>	6601.59	3	100.00	<b>AbortUser</b>	55.70	72397.95	9.46	8	10	1460.23	248.24	36.16
n20-q10-w3-L <sub>1</sub>	15813.95	3443	0.00	OptimalTol	4.12	72574.50	0.04	8	10	4575.30	2241.89	0.00
n20-q10-w3-L <sub>∞</sub>	52730.94	2609	0.00	Optimal	35.79	72453.74	4.34	8	10	1940.03	1008.81	0.00
n20-q10-w1-L <sub>1</sub>	19143.41	2167	0.00	Optimal	1.38	72584.54	0.00	8	10	2430.98	413.27	0.00
n20-q10-w1-L <sub>∞</sub>	86409.91	2173	0.00	OptimalTol	12.74	72511.02	1.20	8	10	1380.59	248.51	0.00

Table 6 continued

Instance	CPLEX							Heuristic				
	TimeIP	Nnodes	Gap (%)	IPStatus	IPobj	Z <sub>1</sub>	Z <sub>2</sub>	#Hubs	#Edges	TimeHB	TimeH	GapH(%)
n20-q15-idealLdBIncng	16.63	0	0.00	OptimalTol	0.00	140945.72	0.00	10	15	2205.45	2029.01	0.00
n20-q15-idealCost	2093.63	1514	0.00	Optimal	73614.00	73614.00	1453.09	10	15	2240.39	291.25	0.00
n20-q15-w9-L <sub>1</sub>	19151.12	2241	0.00	Optimal	14.43	74464.13	44.62	9	15	1510.86	332.39	0.00
n20-q15-w9-L <sub>∞</sub>	56592.53	2532	0.00	Optimal	90.31	74289.61	131.22	9	15	1380.75	1118.41	0.00
n20-q15-w7-L <sub>1</sub>	17730.57	2278	0.00	Optimal	13.92	74806.40	7.37	9	15	2685.70	1879.99	0.00
n20-q15-w7-L <sub>∞</sub>	73827.77	1412	65.00	<b>AbortUser</b>	89.16	74471.65	43.19	9	15	1390.52	987.27	8.31
n20-q15-w5-L <sub>1</sub>	19278.40	2514	0.00	Optimal	10.63	74974.08	1.53	9	15	1770.12	973.57	0.00
n20-q15-w5-L <sub>∞</sub>	102302.62	2576	0.00	OptimalTol	74.78	74620.94	21.73	9	15	2595.63	2180.33	0.00
n20-q15-w3-L <sub>1</sub>	20554.80	4144	0.00	OptimalTol	6.56	75054.60	0.29	9	15	2940.41	441.06	0.00
n20-q15-w3-L <sub>∞</sub>	86973.68	7992	0.00	Optimal	50.65	74750.72	10.51	9	15	2680.38	1554.62	0.00
n20-q15-w1-L <sub>1</sub>	22619.54	2544	0.00	Optimal	2.22	75100.58	0.02	9	15	4825.82	482.58	0.00
n20-q15-w1-L <sub>∞</sub>	202197.22	8504	0.00	OptimalTol	19.30	74913.55	3.12	9	15	1320.80	409.45	0.00
										Avg. CPLEX time in H:		2.273 s
										$\frac{Min}{Max}$ /VarH (C):		0.281/2581.735
										$\frac{Min}{Max}$ /VarH (L):		0.933/388.254
										Avg. $\frac{Min}{Max}$ /VarH ( $\mathcal{L}_1, \mathcal{L}_\infty$ ):		0.917/466.845
										$\frac{Min}{Max}$ /VarH (C):		0.240/2787.974
										$\frac{Min}{Max}$ /VarH (L):		0.957/479.734
										Avg. $\frac{Min}{Max}$ /VarH ( $\mathcal{L}_1, \mathcal{L}_\infty$ ):		0.923/569.789

Table 6 continued

Instance	CPLEX										Heuristic	
	TimeIP	Nnodes	Gap (%)	IPStatus	IPobj	Z <sub>1</sub>	Z <sub>2</sub>	#Hubs	#Edges	TimeH	TimeHB	GapH(%)
n20-q20-idealLdBlncng	18.17	0	0.00	OptimalTol	0.00	160490.00	0.00	10	20	2200.31	1144.16	0.00
n20-q20-idealCost	1543.93	823	0.00	Optimal	76466.97	76466.97	1274.86	10	20	1140.11	912.09	0.00
n20-q20-w9-L <sub>1</sub>	6407.59	3450	0.00	Optimal	12.56	77423.62	29.54	10	20	2320.16	2134.55	0.00
n20-q20-w9-L <sub>∞</sub>	35423.71	785	0.00	Optimal	71.37	77133.25	90.98	10	20	1250.27	425.09	0.00
n20-q20-w7-L <sub>1</sub>	4180.28	868	0.00	Optimal	11.52	77742.13	3.79	10	20	3650.30	2336.19	0.00
n20-q20-w7-L <sub>∞</sub>	46329.46	794	0.00	OptimalTol	77.66	77399.19	33.00	10	20	1680.83	487.44	0.00
n20-q20-w5-L <sub>1</sub>	5687.66	1218	0.00	OptimalTol	8.64	77857.80	0.93	10	20	1175.79	611.41	0.00
n20-q20-w5-L <sub>∞</sub>	9540.10	0	100.00	<b>AbortUser</b>	63.94	77541.48	16.30	10	20	1540.71	939.83	34.31
n20-q20-w3-L <sub>1</sub>	6243.50	1222	0.00	Optimal	5.31	77927.49	0.17	10	20	1020.10	999.70	0.00
n20-q20-w3-L <sub>∞</sub>	195201.59	16	100.00	<b>AbortTimeLim</b>	42.48	77656.69	7.74	10	20	4950.04	1980.02	39.90
n20-q20-w1-L <sub>1</sub>	6059.97	814	0.00	Optimal	1.79	77966.65	0.01	10	20	1455.07	727.53	0.00
n20-q20-w1-L <sub>∞</sub>	31572.73	1095	0.00	Optimal	15.79	77793.93	2.24	10	20	3300.43	1320.17	0.00

Table 6 continued

Instance	CPLEX						Heuristic					
	TimeIP	Nnodes	Gap (%)	IPStatus	IPobj	Z <sub>1</sub>	Z <sub>2</sub>	#Hubs	#Edges	TimeHB	TimeHH	GapH(%)
n20-q25-idealLdBIncng	18.13	0	1.00	OptimalTol	0.00	195584.97	0.00	10	25	2970.56	2732.92	3.041 s
n20-q25-idealCost	2198.38	3392	0.00	Optimal	82247.40	82247.40	1725.18	10	25	3275.18	2423.63	Avg. CPLEX time in H: $\frac{Min}{Max}$ /VarH (C): 0.201/3182.911
n20-q25-w9-L <sub>1</sub>	9522.33	4055	0.00	Optimal	11.85	83463.50	37.75	10	25	2145.66	300.39	$\frac{Min}{Max}$ /VarH (L): 0.977/712.475
n20-q25-w9-L <sub>∞</sub>	60782.922	1213	0.00	OptimalTol	67.33	83095.32	116.16	10	25	2895.59	1360.93	Avg. $\frac{Min}{Max}$ /VarH (L <sub>1</sub> , L <sub>∞</sub> ): 0.945/811.648
n20-q25-w7-L <sub>1</sub>	7567.25	1585	0.00	Optimal	11.02	83896.01	4.84	10	25	2040.76	1244.86	0.00
n20-q25-w7-L <sub>∞</sub>	142069.75	16	100.00	<b>AbortUser</b>	78.54	83519.04	45.16	10	25	2985.11	2029.88	13.77
n20-q25-w5-L <sub>1</sub>	9136.92	2553	0.00	Optimal	8.25	84043.66	1.11	10	25	2275.11	910.05	0.00
n20-q25-w5-L <sub>∞</sub>	189733.98	16	100.00	<b>AbortTimeLim</b>	66.78	83761.02	23.04	10	25	3260.69	1141.24	26.01
n20-q25-w3-L <sub>1</sub>	8705.81	2082	0.00	Optimal	5.06	84126.96	0.21	10	25	4200.52	1596.20	0.00
n20-q25-w3-L <sub>∞</sub>	56783.21	8265	0.00	Optimal	40.59	83780.71	10.00	10	25	1275.02	726.76	0.00
n20-q25-w1-L <sub>1</sub>	6529.61	1321	0.00	Optimal	1.71	84175.47	0.02	10	25	2600.31	1768.21	0.00
n20-q25-w1-L <sub>∞</sub>	46454.45	5653	0.00	Optimal	15.09	83958.12	2.89	10	25	3480.48	3097.62	0.00
												Avg. CPLEX time in H: $\frac{Min}{Max}$ /VarH (C): 0.180/4577.945
												$\frac{Min}{Max}$ /VarH (L): 0.985/1033.684
												Avg. $\frac{Min}{Max}$ /VarH (L <sub>1</sub> , L <sub>∞</sub> ): 0.933/1184.944



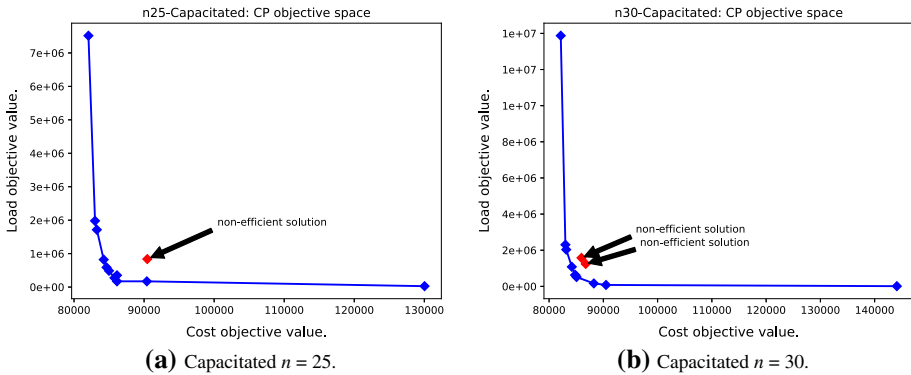


Fig. 12 The CP frontiers corresponding to the CDSHLP instances with  $n = 25, 30$  using the VNS approach

### 5.4 Variable neighborhood methods

Our VNS reports optimal solutions for all instances for which the optimal solutions and objective values are obtained via CPLEX. The only bottleneck of this heuristic approach seems to be its dependency on CPLEX for solving the continuous part of the problem. Therefore, it is very vital for its performance to employ a sophisticated hashing system in order to avoid resolving a problem solved previously.

Due to such a large neighborhood structure, the heuristic was always terminated with the time limit.

For  $n = 25$  and  $n = 30$  (and the values of  $q$  for each instance given in Table 2) for the uncapacitated version we could only use the heuristic approach as CPLEX performed very inefficient. Within the time limit, we have obtained solutions that essentially confirmed the results for aggregated smaller instances (i.e.  $n = 10, 15$  and  $20$ ). The conclusion is therefore that dominance of instance with  $q = 5$  is re-stated.

For the capacitated case, as it can be seen in Fig. 12 and Fig. 13, a smooth boundary has been generated. However, in a couple of cases, the solutions reported by the heuristic (perhaps due to the sub-optimality) did not lie on the smooth curve and are therefore non-efficient solutions.

### 5.5 Random instance with $n = 40$ and $50$

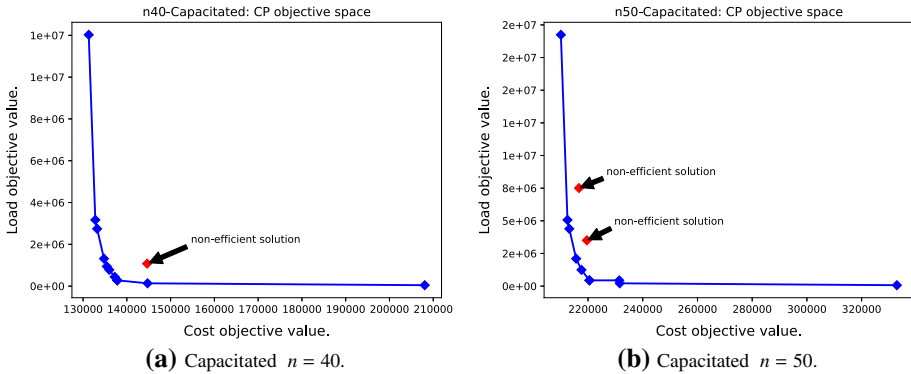
Our algorithm was also tested on a set of random instances without any particular structure. Again for all the cases where the optimal solution was at hand, the heuristic reported that solution.

For the uncapacitated case, given different fixed cost and flow structures, we did not observe any tendency towards installing smaller number of hub edges as we could observe in the real-life data.

In the capacitated case, again we see that for a couple of cases, the solution reported by the heuristic are dominated ones and are not part of the frontier.

## 6 Summary and conclusion

The infrastructure owners (in telecommunication and transportation sectors) usually have a longer term plan and a couple of major objectives. They seek solutions that are close to the



**Fig. 13** The CP frontiers corresponding to the CDSHLP instances with  $n = 40, 50$  using the VNS approach

optimal solution with respect to each objective. Investment in routing and switching centers and the technological connections among them in telecommunications and transforming some platforms to the uni-modal/multi-modal hub nodes and establishing transport corridors are part of the decisions to be made.

In all such applications, the main question is to how design the strategic network such that it has the sufficient capacity and can be resilient and responsive with respect to the current and future demands. This is independent of what operators will be operating the infrastructures.

We have proposed mathematical models for Capacitated and Uncapacitated versions of the problem. These models are intrinsically nonlinear, which are transformed to second order conic counter-parts. The bi-objective model is approached using the Compromise Programming technique. The resulting models are very computationally challenging and hence exact methods have very limited success. We therefore, proposed a very efficient adaptive variable neighborhood search wherein CPLEX is used to solve the resulting subproblems. We then applied the methodology on a case-study from a telecommunication operator that also owns major part of the national infrastructure and obtained practically high quality solutions. The CP has been used to provide a visual illustration of the trade-off between cost and load balance for the problem instances studied.

The future directions include reformulation of the models to models with particular structures to be exploited in exact decomposition techniques, identifying some valid inequalities and proposing some hyper-heuristic approaches and conducting learning-based local search techniques.

**Acknowledgements** This work is supported by the PGMO, the Gaspard Monge Program for Optimisation and Operational Research in the framework of BENMIP project and Interreg 2Seas SPEED project.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## References

Altn, A., Fortz, B., Thorup, M., & Ümit, H. (2009). Intra-domain traffic engineering with shortest path routing protocols. *4OR: A Quarterly Journal of Operations Research*, 7(4), 301–335.

- Alumur, S., & Kara, B. Y. (2008). Network hub location problems: The state of the art. *European Journal of Operational Research*, 190, 1–21.
- Amiri, M., Ekhtiari, M., & Yazdani, M. (2011). Nadir compromise programming: A model for optimization of multi-objective portfolio problem. *Expert Systems with Applications*, 38(6), 7222–7226.
- André, F. J., Cardenete, M. A., & Romero, C. (2008). Using compromise programming for macroeconomic policy making in a general equilibrium framework: Theory and application to the spanish economy. *Journal of the Operational Research Society*, 59(7), 875–883.
- Anton, J. M., & Bielza, C. (2003). Compromise-based approach to road project selection in madrid metropolitan area. *Journal of the Operations Research Society of Japan*, 46(1), 99–122.
- Ballesterro, E. (2007). Compromise programming: A utility-based linear-quadratic composite metric from the trade-off between achievement and balanced (non-corner) solutions. *European Journal of Operational Research*, 182(3), 1369–1382.
- Benders, J. (2005). Partitioning procedures for solving mixed-variables programming problems. *Computational Management Science*, 2(1), 3–19.
- Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1), 238–252.
- Campbell, J. F. (1990). Locating transportation terminals to serve an expanding demand. *Transportation Research Part B: Methodological*, 24(3), 173–192.
- Campbell, J. F., Ernst, A. T., & Krishnamoorthy, M. (2002). Hub location problems. In Z. Drezner & H. W. Hamacher (Eds.), *Facility location: Applications and theory* (pp. 373–407). Berlin: Springer.
- Campbell, J. F., & O’Kelly, M. E. (2012). Twenty-five years of hub location research. *Transportation Science*, 46(2), 153–169.
- Codato, G., & Fischetti, M. (2006). Combinatorial Benders’ cuts for mixed-integer linear programming. *Operations Research-Baltimore then Linthicum*, 54(4), 756.
- Contreras, I., Cordeau, J.-F., & Laporte, G. (2011). The dynamic uncapacitated hub location problem. *Transportation Science*, 45(1), 18–32.
- Correia, I., Nickel, S., & Saldanha-daGama, F. (2014). Multi-product capacitated single-allocation hub location problems: Formulations and inequalities. *Networks and Spatial Economics*, 14(1), 1–25.
- de Camargo, R. S., Miranda, G. Jr., Ferreira, R., & Luna, H. (2009). Multiple allocation hub-and-spoke network design under hub congestion. *Computers & Operations Research*, 36(12), 3097–3106. (new developments on hub location).
- De Camargo, R. S., De Miranda, G., & Ferreira, R. P. (2011). A hybrid outer-approximation/benders decomposition algorithm for the single allocation hub location problem under congestion. *Operations Research Letters*, 39(5), 329–337.
- de Camargo, R. S., & Miranda, G. (2012). Single allocation hub location problem under congestion: Network owner and user perspectives. *Expert Systems with Applications*, 39(3), 3385–3391.
- Elhedhli, S., & Hu, F. (2005a). Hub-and-spoke network design with congestion. *Computers and Operations Research*, 32, 1615–1632.
- Elhedhli, S., & Hu, F. X. (2005b). Hub-and-spoke network design with congestion. *Computers & Operations Research*, 32(6), 1615–1632.
- Elhedhli, S., & Wu, H. (2010). A lagrangean heuristic for hub-and-spoke system design with capacity selection and congestion. *INFORMS Journal on Computing*, 22(2), 282–296.
- Ernst, A., & Krishnamoorthy, M. (1996). Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location Science*, 4(3), 139–154.
- Farahani, R. Z., Hekmatfar, M., Arabani, A. B., & Nikbakhsh, E. (2013). Survey: Hub location problems: A review of models, classification, solution techniques, and applications. *Computers & Industrial Engineering*, 64(4), 1096–1109.
- Fattahi, P., & Fayyaz, S. (2010). A compromise programming model to integrated urban water management. *Water Resources Management*, 24(6), 1211–1227.
- Florentino, H., Irawan, C., Aliano, A., Jones, D., Cantane, D., & Nervis, J. (2017). A multiple objective methodology for sugarcane harvest management with varying maturation periods. *Annals of Operations Research*, 267(1–2), 153–177.
- Gelareh, S. (2008). Hub location models in public transport planning. Ph.D. thesis, Technical University of Kaiserslautern, Germany.
- Gelareh, S., Monemi, R. N., & Nickel, S. (2015). Multi-period hub location problems in transportation. *Transportation Research Part E: Logistics and Transportation Review*, 75, 67–94.
- Gelareh, S., & Nickel, S. (2011). Hub location problems in transportation networks. *Transportation Research Part E: Logistics and Transportation Review*, 47(6), 1092–1111.
- Goldman, A. (1969). Optimal location for centers in a network. *Transportation Science*, 3, 352–360.

- Hakimi, S. L. (1964). Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations Research*, 12, 450–459.
- Hall, R. (1989). Configuration of an overnight package air network. *Transportation Research A*, 23, 139–149.
- Hansen, P., Mladenović, N., & Pérez, J. A. M. (2010). Variable neighbourhood search: Methods and applications. *Annals of Operations Research*, 175(1), 367–407.
- Ilić, A., Urošević, D., Brimberg, J., & Mladenović, N. (2010). A general variable neighborhood search for solving the uncapacitated single allocation  $p$ -hub median problem. *European Journal of Operational Research*, 206(2), 289–300.
- Irawan, C. A., Jones, D., & Ouelhadj, D. (2017). Bi-objective optimisation model for installation scheduling in offshore wind farms. *Computers & Operations Research*, 78, 393–407.
- Jones, D., & Tamiz, M. (2010). *Practical goal programming*. Berlin: Springer.
- Kanellopoulos, A., Gerdessen, J., & Claassen, G. (2015). Compromise programming: Non-interactive calibration of utility-based metrics. *European Journal of Operational Research*, 244(2), 519–524.
- Kara, B. Y., & Taner, M. R. (2011). Hub location problems: The location of interacting facilities. In H. A. Eiselt & V. Marianov (Eds.), *Foundations of location analysis* (pp. 273–288). Berlin: Springer.
- Liberatore, F., Ortuño, M. T., Tirado, G., Vitoriano, B., & Scaparra, M. P. (2014). A hierarchical compromise model for the joint optimization of recovery operations and distribution of emergency goods in humanitarian logistics. *Computers & Operations Research*, 42, 3–13.
- Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research*, 24(11), 1097–1100.
- Mladenović, N., Todosijević, R., & Urošević, D. (2016). Less is more: Basic variable neighborhood search for minimum differential dispersion problem. *Information Sciences*, 326, 160–171.
- Monemi, R. N., Gelareh, S., Hanafi, S., & Maculan, N. (2017). A co-opetitive framework for the hub location problems in transportation networks. *Optimization*, 66(12), 2089–2106.
- O’Kelly, M. (1986a). The location of interacting hub facilities. *Transportation Science*, 20(2), 92–105.
- O’Kelly, M., & Lao, Y. (1991). Mode choice in a hub-and-spoke network: A zero-one linear programming approach. *Geographical Analysis*, 23, 283–297.
- O’Kelly, M. E. (1986b). Activity levels at hub facilities in interacting networks. *Geographical Analysis*, 18(4), 343–356.
- 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., & Bryan, D. L. (1998). Hub location with flow economies of scale. *Transportation Research Part B: Methodological*, 32(8), 605–616.
- Rahimi, Y., Tavakkoli-Moghaddam, R., Mohammadi, M., & Sadeghi, M. (2016). Multi-objective hub network design under uncertainty considering congestion: An m/m/c/k queue system. *Applied Mathematical Modelling*, 40(5), 4179–4198.
- Rodríguez-Martín, I., Salazar-González, J.-J., & Yaman, H. (2014). A branch-and-cut algorithm for the hub location and routing problem. *Computers & Operations Research*, 50, 161–174.
- Romero, C., & Rehman, T. (2003). *Multiple criteria analysis for agricultural decisions* (Vol. 11). Amsterdam: Elsevier.
- Romero, C., Tamiz, M., & Jones, D. (1998). Goal programming, compromise programming and reference point method formulations: Linkages and utility interpretations. *Journal of the Operational Research Society*, 49(9), 986–991.
- Sasaki, M., Campbell, J. F., Krishnamoorthy, M., & Ernst, A. T. (2014). A stackelberg hub arc location model for a competitive environment. *Computers & Operations Research*, 47, 27–41.
- Todosijević, R., Urošević, D., Mladenović, N., & Hanafi, S. (2015). A general variable neighborhood search for solving the uncapacitated  $r$ -allocation  $p$ -hub median problem. *Optimization Letters*, <https://doi.org/10.1007/s11590-015-0867-6>.
- Zeleny, M. (1973). Compromise programming. In *Multiple criteria decision making* (p. 286).