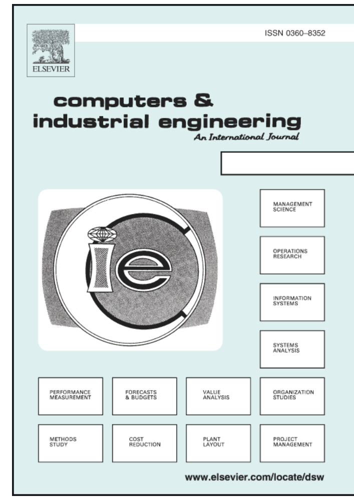
Accepted Manuscript



A chance constrained programming approach for uncertain *p*-hub center location problem

Yuan Gao, Zhongfeng Qin

PII: S0360-8352(16)30357-6
DOI: http://dx.doi.org/10.1016/j.cie.2016.09.017
Reference: CAIE 4473

To appear in: *Computers & Industrial Engineering*

Received Date: 19 December 2015
Accepted Date: 16 September 2016

A chance constrained programming approach for uncertain p-hub center location problem

Yuan Gao[1], Zhongfeng Qin[2]

[1]*State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China*

[2]*School of Economics and Management, Beihang University, Beijing 100191, China*

gaoyuan@bjtu.edu.cn, qin@buaa.edu.cn

Abstract

The p-hub center location problem aims to locate p hubs and allocate other nodes to these hub nodes in order to minimize the maximal travel time. It is more important for time-sensitive distribution systems. Due to the presence of uncertainty, more researches are recently focused on the problem in non-deterministic environment. This paper joins the research stream by considering travel times as uncertain variables instead of random variables or fuzzy ones. The goal is to model the p-hub center problem based on experts' subjective belief in the case of lack of data. The uncertain distribution of the maximal travel time is first derived and then a chance constrained programming model is formulated. The deterministic equivalent forms are further given when the information of uncertainty distributions is provided. A hybrid intelligent algorithm is designed to solve the proposed models and numerical examples are presented to illustrate the application of this approach and the effectiveness of the algorithm.

Keywords: uncertainty modelling; p-hub center location problem; uncertain variable; uncertain measure; chance-constrained programming.

1 Introduction

Hubs are, in practice, commonly used in transportation, logistics, telecommunication systems and serves as consolidation, switching and sorting centers in a complex system (Campbell et al., 2007). The p-hub center location problem involves the location of hub facilities and the routing from origins to destinations to minimize the maximal travel time (or cost, distance, etc.) between any origin-destination pair. The problem is useful for time-sensitive distribution systems such as emergency services, express mail service or timely delivery of perishable products (Campbell et al., 2007). Therefore, since it was initialized by Campbell (1994), more research interests are draw to study and extend the problem. The latest review can be found in Compbell and O'Kelly (2012) and Farahani et al. (2013).

In general, hub location problems are strategic in nature, which implies that the travel times may change with time. Therefore, it is meaningful to consider the problem within an uncertain environment. One main stream of research is to deal with uncertainty as randomness, i.e., stochastic p-hub center location problem. Sim et al. (2009) was first one to present stochastic p-hub center problem and established a chance-constrained programming with service-level constraints. Then Yang et al. (2011) extended the problem by assuming discrete random travel time, and Hult et al. (2013) developed exact solution approaches based on variable reduction and a separation algorithm to solve an uncapacitated single allocation case.

Another stream of research is to study p-hub center location problem in fuzzy environment. For instance, Yang et al. (2013a) first proposed a fuzzy p-hub center problem in which the travel times are characterized by normal fuzzy vectors. Based on the same setting, Yang et al. (2013b) further presented a risk aversion formulation by adopting value-at-risk criterion in the objective function. By using the criterion, Yang et al. (2014) recently developed a robust optimization method to describe travel times by employing parametric possibility distributions. Similarly, Bashiri et al. (2013) considered a hybrid approach to the capacitated case with fuzzy data and employed genetic algorithm to solve the problem.

In practice, there are often lack of data about future changes or the implementation will take considerable time or cost. A more feasible and economic way is to estimate the parameters by experts based on their subjective information and experiences. Liu (2010) proposed uncertainty theory to describe such a nondeterministic phenomena. Since then, uncertainty theory has been applied to practical problems such as Chen et al. (2012) and Gao et al. (2015) as a new approach dealing with uncertainty. Gao (2011) introduced the shortest path problem and gave the uncertainty distribution of the shortest path length, and Gao (2012) proposed single facility location model in which products assignment is evaluated by the concept of satisfaction degree. Wen et al. (2014) presented chance-constrained formulation for capacitated facility location-allocation problem when demands are uncertain. Han et al. (2014) studied the maximum flow in an uncertain network in which arc capacities are uncertain variables.

This paper focuses on the single allocation p-hub center location problem with lack of data about travel times. Instead of estimating the travel times by statistical methods, this paper assume that these quantities are evaluated by domain experts. Specifically, in the framework of uncertainty theory, this paper regards the travel times between the origins and destinations as uncertain variables and propose a new formulation for the problem. Chance constrained programming approach is employed to model the problem and further formulate an uncertain optimization model for decision makers. The proposed model is then transformed to deterministic equivalent forms when uncertainty distributions are provided. In order to solve these models, we design a hybrid intelligent algorithm by combining principle of nearby into genetic algorithm.

The rest of the paper is organized as follows. In Section 2, we first formulate a chance constrained programming model and then discuss its deterministic equivalent forms. Sections 3 provides a heuristic solution procedure to solve the proposed models. In Section 4, numerical examples are given to illustrate the application and effectiveness of the proposed models. A brief conclusion is given in Section 5 and finally some necessary preliminaries are given in Appendix.

2 Uncertain p-hub Center Problem

This section is divided into three subsections. The first subsection describes the p-hub center location problem in uncertain environment, the second one formulates a chance constrained programming model, and the third one discusses the deterministic equivalent forms of the proposed model.

2.1 Problem Description

Assume that there are n nodes in the network and the number of hubs to locate is given exogenously and denoted by p. Let η_{ij} be the travel time on the link from node i to j, which is considered as an uncertain variable defined on the uncertainty space $(\Theta, \mathcal{P}, \mathcal{M})$. A path $i \to k \to m \to j$ represents a unit of demand originating at node i destined for j traveling through hub k first then hub m. If α is a discount factor denoting economies of scale on the inter-hub linkage, then the total travel time on this path is $\eta_{ik} + \alpha\eta_{km} + \eta_{mj}$ which is also an uncertain variable. Note that $k = m$ implies that only one hub is used and naturally the discount vanishes.

The decision variable x_{ik} is introduced as a binary variable to represent the assignment of node i to hub k for $i, k = 1, 2, \cdots, n$. Here $x_{kk} = 1$ indicates that node k is assigned to itself and it is actually a hub node. We assume that all of hub nodes are connected to one another, however, any two non-hub nodes are never connected directly. Moreover, each non-hub node is assigned to a single hub. In addition, there is no cost for setting up hubs and there are no capacity limits.

If all the travel time η_{ij} are all known in advance, i.e., deterministic values, the uncertain p-hub center location problem becomes to a traditional deterministic one, the aim of which is to minimize the maximal travel time. We denote the travel time by w_{ij} in this part. Typically, the deterministic p-hub center problem can be formulated as

$$\begin{cases} \min\limits_{\boldsymbol{x}} & f(\boldsymbol{w}, \boldsymbol{x}) = \max\limits_{i,j,k,m} \{(w_{ik} + \alpha w_{km} + w_{mj}) x_{ik} x_{jm}\} \\ \text{s.t.} & \sum_{k=1}^{n} x_{kk} = p \quad (i) \\ & \sum_{k=1}^{n} x_{ik} = 1, \ i = 1, 2, \cdots, n \quad (ii) \\ & x_{ik} \le x_{kk}, \ i, k = 1, 2, \cdots, n \quad (iii) \\ & x_{ik} = \{0, 1\}, \ i, k = 1, 2, \cdots, n \quad (iv) \end{cases} \tag{1}$$

where $\boldsymbol{w} = (w_{ij})$ and $\boldsymbol{x} = (x_{ij})$, $i, j = 1, \cdots, n$.

For model (1), the objective function $f(\boldsymbol{w}, \boldsymbol{x})$ represents the maximal travel time between two nodes in the p-hub location $\boldsymbol{x}$. Constraint (i) stipulates that exactly p hub nodes are chosen and constraints (ii) stipulate that non-hub node i is assigned to precisely one hub node. Constraints (iii) enforce that node i is assigned to a hub node at k only if a hub is located at node k. Finally, constraints (iv) define the decision variable types. For sake of simplicity, we write

$$X = \{\boldsymbol{x} | \boldsymbol{x} \text{ satisfies constraints } (i), (ii), (iii) \text{ and } (iv) \text{ of model } (1)\},$$

which is the collection of all feasible solutions of p-hub center location problem.

Denote by $\boldsymbol{x}^*$ the optimal solution of model (1). Then the optimal objective value of model (1) can be expressed as follows

$$F(\boldsymbol{w}) = f(\boldsymbol{w}, \boldsymbol{x}^*) = \min_{\boldsymbol{x} \in X} f(\boldsymbol{w}, \boldsymbol{x}). \tag{2}$$

In other words, considering all the feasible solutions, $F(\boldsymbol{w})$ represents the minimum value of maximal travel time. Furthermore, $F(\boldsymbol{w})$ is determined by the number of nodes n, the number of hubs p and link travel times $\boldsymbol{w}$. It is easy to verify that $F(\boldsymbol{w})$ is a strictly increasing function with regard to w_{ij}, $i, j = 1, \cdots, n$. More precisely, for given $\boldsymbol{w} = (w_{ij})$ and $\boldsymbol{w}' = (w'_{ij})$, $F(\boldsymbol{w})$ has the following monotonicity,
(1) $F(\boldsymbol{w}) \le F(\boldsymbol{w}')$ when $w_{ij} \le w'_{ij}$ for all $i, j = 1, \cdots, n$;
(2) $F(\boldsymbol{w}) < F(\boldsymbol{w}')$ when $w_{ij} < w'_{ij}$ for all $i, j = 1, \cdots, n$.
Generally speaking, if some links' travel times increase, then the maximal travel time will not necessarily increase. However, if all the travel times increase, then it is certain that the maximal travel time will increase. As a result, the monotonicity of $F(\boldsymbol{w})$ is a natural thing.

If the travel times are not deterministic values, i.e., $\boldsymbol{w} = (w_{ij})$ being replaced with uncertain travel times $\boldsymbol{\eta} = (\eta_{ij})$, $i, j = 1, 2, \cdots, n$, then model (1) will not work. This is because the travel time of path $i \to k \to m \to j$, namely, $(\eta_{ik} + \alpha \eta_{km} + \eta_{mj})$, is an uncertain variable, and we can not compare uncertain variables in a similar way to compare deterministic ones. Moreover, we may not find a p-hub location $\boldsymbol{x}^*$

which has a minimum value of the maximal travel time in all situations. In order to modify model (1) in uncertain environment, next we first consider the uncertainty distribution of the maximal travel time.

Since in the uncertain environment η_{ij} may take many different values, the minimum value of maximal travel time $F(\boldsymbol{\eta})$ may naturally take different values. As a result, a more meaningful way is to investigate the uncertainty distribution of $F(\boldsymbol{\eta})$, which is denoted by $\Psi(t)$, i.e.

$$\Psi(t) = \mathcal{M}\{F(\boldsymbol{\eta}) \leq t\},$$

where t is a positive real number.

Assume that η_{ij} has a regular uncertainty distribution $\Phi_{ij}(t)$ for $i, j = 1, \cdots, n$. Since F is a strictly increasing function, according to the operational law of uncertain variables (Lemma 1 in Appendix), the inverse uncertainty distribution function $\Psi^{-1}(\beta)$ of $F(\eta)$ can be calculated by

$$\Psi^{-1}(\beta) = F(\boldsymbol{\Phi}_{\beta}^{-1}), \quad \beta \in (0, 1)$$

where $\boldsymbol{\Phi}_{\beta}^{-1} = (\Phi_{ij}^{-1}(\beta))$, $i, j = 1, \cdots, n$. According to formula (2), $\Psi^{-1}(\beta)$ can be specifically expressed as

$$\Psi^{-1}(\beta) = \min_{\boldsymbol{x} \in X} f(\boldsymbol{\Phi}_{\beta}^{-1}, \boldsymbol{x}). \tag{3}$$

Obviously, $\Psi^{-1}(\beta)$ is just the optimal value of model (1) with $w_{ij} = \Phi_{ij}^{-1}(\beta)$, $i, j = 1, \cdots, n$. Via formula (3), we can numerically obtain $\Psi^{-1}(\beta), \beta \in (0, 1)$, which is the inverse function of $\Psi(t)$.

2.2 Formulation of Chance Constrained Programming

In the uncertain environment, the decision makers sometimes need to determine a time level $\bar{T}$, such that there exists a p-hub location $\boldsymbol{x}$ satisfying $\mathcal{M}\{f(\boldsymbol{\eta}, \boldsymbol{x}) \leq \bar{T}\} \geq \beta$ where β is a predetermined chance level. For instance, given $\beta = 0.9$, the decision makers have to determine a time level $\bar{T}$ and then choose a p-hub location $\boldsymbol{x}$ satisfying $\mathcal{M}\{f(\boldsymbol{\eta}, \boldsymbol{x}) \leq \bar{T}\} \geq 0.9$. That is to say, if the decision makers choose the p-hub location $\boldsymbol{x}$, then the travel time between any two nodes will be lower than $\bar{T}$ with a chance of at least 90%. Of course, for a given chance level β, the decision makers always wish to determine a sufficiently low time level $\bar{T}$.

With that in mind, Charnes and Cooper (1959) proposed the method of chance constrained programming, which has been developed as a major approach to model uncertain decision systems. To further investigate the uncertain models, we first give a concept of β-optimal p-hub location.

Definition 1. *For a p-hub center location problem with uncertain link travel time $\boldsymbol{\eta}$, given chance level β, the feasible solution $\boldsymbol{x}^*$ is called the β-optimal p-hub location, if for any feasible solution $\boldsymbol{x}$, the solution $\boldsymbol{x}^*$ satisfies*

$$\bar{T} = \min\Big\{T | \mathcal{M}\{f(\boldsymbol{\eta}, \boldsymbol{x}^*) \leq T\} \geq \beta\Big\} \leq \min\Big\{T | \mathcal{M}\{f(\boldsymbol{\eta}, \boldsymbol{x}) \leq T\} \geq \beta\Big\},$$

where $f(\boldsymbol{\eta}, \boldsymbol{x}) = \max\limits_{i,j,k,m} \{(\eta_{ik} + \alpha\eta_{km} + \eta_{mj}) x_{ik} x_{jm}\}$.

Based on the idea of Definition 1, the following uncertain chance constrained programming (UCCP) for the p-hub center location problem is meaningful and suitable, *i.e.*,

$$(\text{UCCP}) \begin{cases} \min & \bar{T} \\ \text{s.t.} & \mathcal{M}\{f(\boldsymbol{\eta}, \boldsymbol{x}) \leq \bar{T}\} \geq \beta, \qquad (v) \\ & \boldsymbol{x} \in X, \end{cases} \tag{4}$$

It is easy to see that the optimal solution of model (4) is just the β-optimal p-hub location.

2.3 Deterministic Equivalent Forms

Similar to stochastic situation (Sim et al., 2009), we assume that the travel time on each link is independent of those on all other links of the network. The sense of independence is given in Definition 3 in the framework of uncertainty theory. Then we present the deterministic equivalent forms of Model UCCP when there are more information about the uncertainty distributions of the travel times.

Theorem 1. *Suppose that travel times η_{ij} $(i, j = 1, 2, \cdots, n)$ are mutually independent uncertain variables with regular uncertainty distributions Φ_{ij}, respectively. Then Model UCCP is equivalent to the following deterministic integer programming model,*

$$\begin{cases} \min & \bar{T} \\ s.t. & [\Phi_{ik}^{-1}(\beta) + \alpha\Phi_{km}^{-1}(\beta) + \Phi_{mj}^{-1}(\beta)]x_{ik}x_{jm} \le \bar{T},\ i, j, k, m = 1, 2, \cdots, n \\ & \boldsymbol{x} \in X. \end{cases} \tag{5}$$

Proof: Note that for any $\boldsymbol{x} \in X$, $f(\boldsymbol{\eta}, \boldsymbol{x}) = \max\limits_{i,j,k,m} \{(\eta_{ik} + \alpha\eta_{km} + \eta_{mj})x_{ik}x_{jm}\}$ is a strictly increasing function with η_{ij}, $i, j = 1, 2, \cdots, n$. Denote by $\Psi_{\boldsymbol{x}}(t)$ the uncertainty distribution of $f(\boldsymbol{\eta}, \boldsymbol{x})$. According to Lemma 1, $\Psi_{\boldsymbol{x}}(t)$ has an inverse uncertainty distribution

$$\Psi_{\boldsymbol{x}}^{-1}(\beta) = f(\boldsymbol{\Phi}_{\beta}^{-1}, \boldsymbol{x}) = \max_{i,j,k,m} \Big\{(\Phi_{ik}^{-1}(\beta) + \alpha\Phi_{km}^{-1}(\beta) + \Phi_{mj}^{-1}(\beta))x_{ik}x_{jm}\Big\}, \tag{6}$$

where $\boldsymbol{\Phi}_{\beta}^{-1} = (\Phi_{ij}^{-1}(\beta))$, $i, j = 1, 2, \cdots, n$. According to the definition of uncertainty distribution, the constraint (v) of Model UCCP can be rewritten as

$$\Psi_x(\bar{T}) = \mathcal{M}\{f(\boldsymbol{\eta}, \boldsymbol{x}) \le \bar{T}\} \ge \beta. \tag{7}$$

It follows from the definition of regular distribution (Definition 2 in Appendix) that $\Psi_{\boldsymbol{x}}(\Psi_{\boldsymbol{x}}^{-1}(\beta)) = \beta$. Since the uncertainty distribution $\Psi_{\boldsymbol{x}}(t)$ is an increasing function with t, formula (7) can be reformulated as

$$\Psi_x^{-1}(\beta) \le \bar{T}.$$

That is,

$$\max_{i,j,k,m} \Big\{(\Phi_{ik}^{-1}(\beta) + \alpha\Phi_{km}^{-1}(\beta) + \Phi_{mj}^{-1}(\beta))x_{ik}x_{jm}\Big\} \le \bar{T},$$

which is equivalent to

$$[\Phi_{ik}^{-1}(\beta) + \alpha\Phi_{km}^{-1}(\beta) + \Phi_{mj}^{-1}(\beta)]x_{ik}x_{jm} \le \bar{T}, \qquad i, j, k, m = 1, 2, \cdots, n.$$

The proof is completed. □

Theorem 1 states that as long as the uncertainty distribution of each travel time is obtained, Model UCPP can be converted into a deterministic mathematical programming. Further, if these travel times have the same type of uncertainty distributions, then we can obtain a simpler form. Next three corollaries are given when all the uncertainty distributions are linear, zigzag and normal, respectively.

Corollary 1. *If the travel times $\eta_{ik} = \mathcal{L}(a_{ik}, b_{ik}), \eta_{km} = \mathcal{L}(a_{km}, b_{km})$ and $\eta_{mj} = \mathcal{L}(a_{mj}, b_{mj})$ are mutually independent linear uncertain variables, then Model UCPP is equivalent to the following form,*

$$\begin{cases} \min & \bar{T} \\ s.t. & \bar{T} \ge (1-\beta)a_{ij}^{km} + \beta b_{ij}^{km},\ i, j, k, m = 1, 2, \cdots, n \\ & (a_{ik} + \alpha a_{km} + a_{mj})x_{ik}x_{jm} = a_{ij}^{km},\ i, j, k, m = 1, 2, \cdots, n \\ & (b_{ik} + \alpha b_{km} + b_{mj})x_{ik}x_{jm} = b_{ij}^{km},\ i, j, k, m = 1, 2, \cdots, n \\ & \boldsymbol{x} \in X. \end{cases} \tag{8}$$

Corollary 2. *If the travel times* $\eta_{ik} = \mathcal{Z}(a_{ik}, b_{ik}, c_{ik}), \eta_{km} = \mathcal{Z}(a_{km}, b_{km}, c_{km})$ *and* $\eta_{mj} = \mathcal{Z}(a_{mj}, b_{mj}, c_{mj})$ *are mutually independent zigzag uncertain variables, then when* $0 < \beta \leq 0.5$*, Model UCPP is equivalent to the following form,*

$$\begin{cases} \min & \bar{T} \\ s.t. & \bar{T} \geq (1-2\beta)a_{ij}^{km} + 2\beta b_{ij}^{km},\ i,j,k,m = 1,2,\cdots,n \\ & (a_{ik} + \alpha a_{km} + a_{mj})x_{ik}x_{jm} = a_{ij}^{km},\ i,j,k,m = 1,2,\cdots,n \\ & (b_{ik} + \alpha b_{km} + b_{mj})x_{ik}x_{jm} = b_{ij}^{km},\ i,j,k,m = 1,2,\cdots,n \\ & \boldsymbol{x} \in X. \end{cases} \tag{9}$$

and when $0.5 \leq \beta \leq 1$*, Model UCPP is equivalent to the following one,*

$$\begin{cases} \min & \bar{T} \\ s.t. & \bar{T} \geq (2-2\beta)b_{ij}^{km} + (2\beta-1)c_{ij}^{km},\ i,j,k,m = 1,2,\cdots,n \\ & (b_{ik} + \alpha b_{km} + b_{mj})x_{ik}x_{jm} = b_{ij}^{km},\ i,j,k,m = 1,2,\cdots,n \\ & (c_{ik} + \alpha c_{km} + c_{mj})x_{ik}x_{jm} = c_{ij}^{km},\ i,j,k,m = 1,2,\cdots,n \\ & \boldsymbol{x} \in X. \end{cases} \tag{10}$$

Corollary 3. *If the travel times* $\eta_{ik} = \mathcal{N}(e_{ik}, \sigma_{ik}), \eta_{km} = \mathcal{N}(e_{km}, \sigma_{km})$ *and* $\eta_{mj} = \mathcal{N}(e_{mj}, \sigma_{mj})$ *are mutually independent normal uncertain variables, then Model UCPP is equivalent to the following form,*

$$\begin{cases} \min & \bar{T} \\ s.t. & \bar{T} \geq e_{ij}^{km} + \frac{\sqrt{3}}{\pi}\ln\left(\frac{\beta}{1-\beta}\right)\sigma_{ij}^{km},\ i,j,k,m = 1,2,\cdots,n \\ & (e_{ik} + \alpha e_{km} + e_{mj})x_{ik}x_{jm} = e_{ij}^{km},\ i,j,k,m = 1,2,\cdots,n \\ & (\sigma_{ik} + \alpha\sigma_{km} + \sigma_{mj})x_{ik}x_{jm} = \sigma_{ij}^{km},\ i,j,k,m = 1,2,\cdots,n \\ & \boldsymbol{x} \in X. \end{cases} \tag{11}$$

Assume that $\boldsymbol{x}^*$ is the optimal solution to model (5) and $\bar{T}^*$ is the corresponding optimal objective value. The following equation always holds,

$$\bar{T}^* = \max_{i,j,k,m}\left\{(\Phi_{ik}^{-1}(\beta) + \alpha\Phi_{km}^{-1}(\beta) + \Phi_{mj}^{-1}(\beta))x_{ik}^*x_{jm}^*\right\}.$$

Consequently, we may obtain the following conclusion.

Theorem 2. *The optimal solution of Model UCCP is just that of the following model*

$$\begin{cases} \min & f(\boldsymbol{\Phi}_\beta^{-1}, \boldsymbol{x}) = \max_{i,j,k,m}\left\{(\Phi_{ik}^{-1}(\beta) + \alpha\Phi_{km}^{-1}(\beta) + \Phi_{mj}^{-1}(\beta))x_{ik}x_{jm}\right\} \\ s.t. & \boldsymbol{x} \in X. \end{cases} \tag{12}$$

Remark 1. *Comparing with model (1), Theorem 2 states that given a chance level* β*, model (12) equivalently presents a deterministic p-hub center location problem with n nodes and the link travel times are* $\boldsymbol{\Phi}_\beta^{-1} = (\Phi_{ij}^{-1}(\beta))$*,* $i,j = 1,2,\cdots,n$*.*

Remark 2. *Comparing with formula (3), it can be observed that for any given chance level* β*, the optimal objective value of Model UCCP is just equal to* $\Psi^{-1}(\beta)$*, in which* Ψ^{-1} *is the inverse uncertainty distribution of* $F(\boldsymbol{\eta})$*.*

3 Solution Procedure

As stated in the above section, to solve the Model UCCP, we can first transform it into a deterministic p-hub center location model based on the given uncertainty distributions. In this section, we will propose a hybrid intelligent algorithm, which combines the principle of nearby and the genetic algorithm (GA), to solve the transformed optimization models.

3.1 Principle of Nearby

In order to improve the efficiency of the algorithm, we propose a principle of nearby which means that each non-hub node is assigned to the hub node that is nearest to it. Actually, this principle is widely adopted in practice, no matter in location problems or not. In general, the item that is nearest naturally comes first in the practical decisions. However, it is worth pointing out that the principle of nearby may not ensure that a feasible solution is optimal. We consider the example shown in Figure 1 in which nodes a, b and c are hub nodes. In Case 1, the assignment of non-hub nodes follows the principle of nearby, and the longest path is $e \to b \to c \to f$ whose travel time is 13. In Case 2, the longest path is $e \to a \to f$ whose travel time is 12. Although sometimes it does not provide an optimal solution, it does provide a pretty good one, which is closer to the optimal solution.

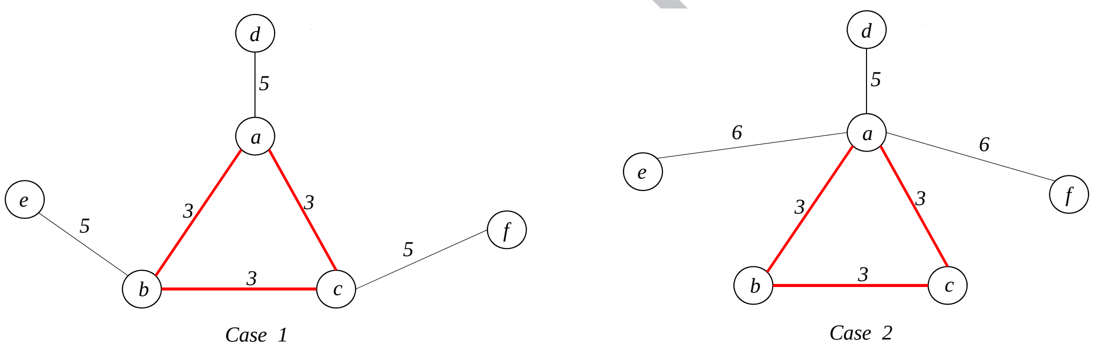


Figure 1: An illustration of principle of nearby.

We only adopt this principle in the process of the assignment of non-hub nodes, i.e., to assign non-hub nodes to hub nodes in the process of initialization and crossover of GA. Since the principle may lead to non-optimal solutions, we do not use it to generate all solutions in the process of the algorithm. To be more precise, we set a probability level P_{pn} in advance. Once an assignment for non-hub node is needed, a random number $r \in (0, 1)$ is first generated. If $r < P_{pn}$, then the principle of nearby is adopted to generate an assignment for the non-hub node. Otherwise, the assignment is randomly generated.

3.2 Hybrid Intelligent Algorithm

In this part, we propose a hybrid intelligent algorithm by combining the principle of nearby into the genetic algorithm. Genetic algorithm (Holland, 1975) is a stochastic search method based on the mechanics of natural selection and natural genetics. It has the characteristics of searching the optimal solution globally and is easy to be combined with other algorithms. As a result, GA has achieved considerable success in providing optimal or near-optimal solutions to many complex optimization problems (Gen and Cheng, 2000) since it was proposed. A typical GA first requires a representation of the solution and a fitness function to evaluate the solutions, and then is implemented by the process of initialization, selection, crossover and mutation operations. The key procedures and genetic operations of the hybrid intelligent algorithm is stated as follows.

Coding Rule: In the p-hub center location problem, we use a $2n$-dimensional array to represent a chromosome (*i.e.*, a solution). The array consists of two parts: Hub-Array and Assign-Array. The former one contains n bits and each bit indicates whether the corresponding node is a hub. In other words, if the value of the kth bit is 1, then node k is a hub node; otherwise, the node k is a non-hub node. The latter one also contains n bits and the value in each bit position indicates the number of hub location that the corresponding node is assigned to. It is worth pointing out that each hub node is assigned to itself in the Assign-Array.

This is an effective representation since it can guarantee the feasibility of each chromosome. It was used first by Topcuoglu et al. (2005) for deterministic hub location problem and then by Yang et al. (2013b) for fuzzy hub center problem. Figure 2 illustrates a possible structure of a chromosome for the problem with 10 nodes and 3 hubs. In this figure, nodes 2, 6 and 9 are chosen as hubs. Hence, the corresponding bits of

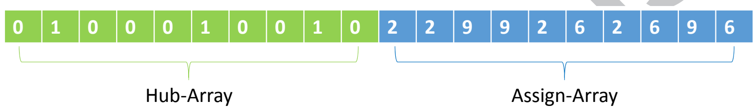


Figure 2: A potential code of a chromosome with 10 nodes and 3 hubs.

the chromosome are set equal to 1 and 0 otherwise. In the Assign-Array, the first bit is “2”, which implies that non-hub node 1 is assigned to hub node 2. In addition, “2” in the second bit indicates that hub node 2 is assigned to itself.

Initialization Process: In this work, we assume that the number of hubs is exogenously given as p. Denote by *pop_size* the population size. In the initialization process, we need to generate *pop_size* chromosomes to construct an initial population. The following process is repeated for *pop_size* times to obtain the first generation and each time will generate a feasible chromosome.

Step 1: Generate a Hub-Array by randomly choosing p different nodes and set the value of bits in the Hub-Array following the Coding Rule.

Step 2: Generate a random number $r \in (0, 1)$. If $r < P_{pn}$, then assign non-hub nodes according to the principle of nearby and then generate the Assign-Array following the Coding Rule. If $r \geq P_{pn}$, then generate the Assign-Array by randomly assigning each non-hub node to only single hub node.

Crossover Operation: Let P_c be the probability level for the selection of parent chromosomes to be crossed over. Typically, a real number $r_c \in (0, 1)$ is randomly generated first. If $r_c < P_c$, randomly select two chromosomes i and j as the parents to crossover. Two offsprings i' and j' are produced according to the following procedure (See Figure 3):

Step 1: Construct a set H of hub nodes by merging two sets of hub nodes represented by two parent chromosomes.

Step 2: Randomly select hub nodes from H to produce two new Hub-Arrays which are as the hub nodes of two offsprings i' and j'.

Step 3: Generate a random number $r \in (0, 1)$. If $r < P_{pn}$, Assign-Arrays of two offsprings i' and j' are respectively generated following the principle of nearby. If $r \geq P_{pn}$, Assign-Arrays of offsprings i' and j' are respectively generated by randomly assigning each non-hub node to only single hub node.

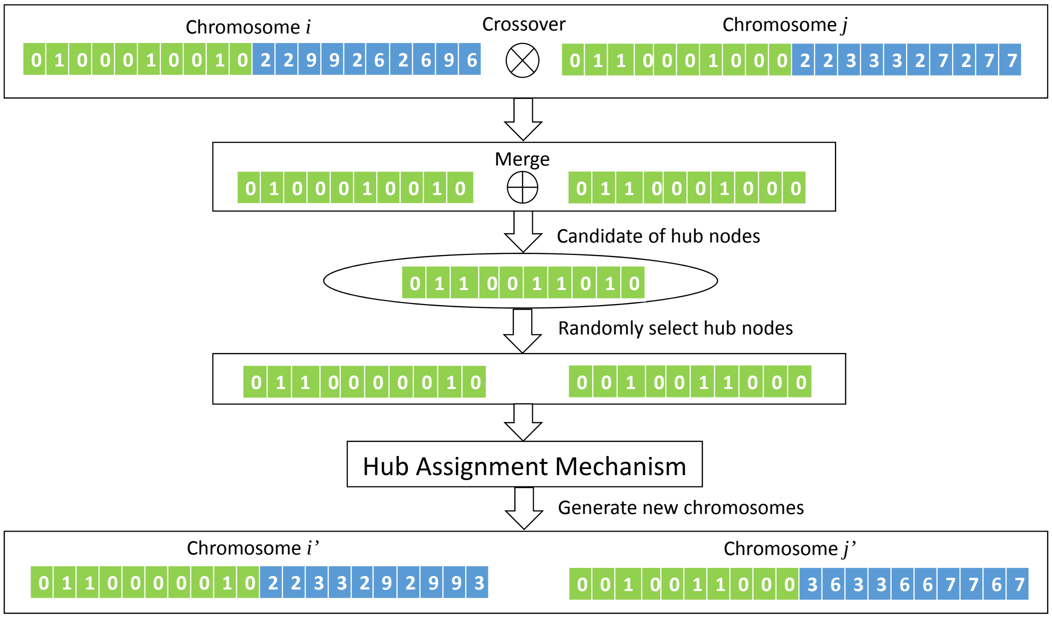


Figure 3: An illustration of crossover operation

The above crossover operation is repeated for $pop\_size/2$ times. Next we take Figure 3 as an example to interpret the whole process. Note that the nodes 2, 6 and 9 are hubs for chromosomes i and nodes 2, 3 and 7 are hubs for chromosomes j. Then, the union of hub sets is $H = \{2, 3, 6, 7, 9\}$, which is the candidate set of hub nodes for the offsprings. In other words, two offsprings have to choose the nodes from H as hubs. In this figure, two new Hub-Arrays are produced, and their hub nodes are $\{2, 3, 9\}$ and $\{3, 6, 7\}$, respectively. Further, two Assign-Arrays are regenerated following the hub assignment mechanism stated in Step 3. Finally, two new chromosomes i' and j' are obtained by the crossover operation.

Mutation Operation: Two types of mutation operations are adopted in our algorithm, namely hub node mutation and non-hub node mutation. The corresponding probability levels of these two mutations are set P_m and P'_m, respectively.

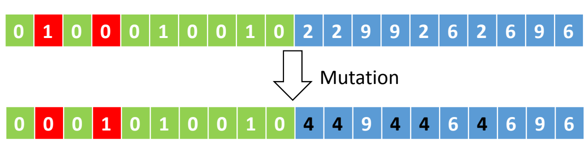


Figure 4: An illustration of hub node mutation

For the hub node mutation, one hub node and one non-hub node of one chromosome are randomly chosen, the positions of which are then exchanged. Specifically, the original hub node becomes a non-hub node, and meanwhile the original non-hub node becomes a hub node. Then, the non-hub nodes that were assigned to original hub node are reassigned to the new hub node. In the example given by Figure 4, node 2 can be replaced by node 4 as a hub node in order to avoid repetition and maintain a balance of number of assigned nodes to hubs. After the mutation, the non-hub nodes 1, 2, 5 and 7 that were assigned to node 2 are now reassigned to the new hub node 4. Naturally, node 4 is also reassigned to itself.

For the hub node mutation, we will keep the hub nodes unchanged and randomly choose two non hub nodes that are assigned to different hubs. Then we exchange these two non-hub nodes. An illustration in Figure 5 shows that non-hub nodes 1 and 8 are chosen then reassigned to hub nodes 6 and 2, respectively.

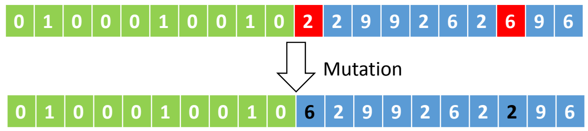


Figure 5: An illustration of non-hub node mutation

In our algorithm, the mutation operation is processed in the following two steps,

Step 1: Randomly generate a real number $r_1 \in (0,1)$. If $r_1 < P_m$, then carry out the hub node mutation. Otherwise, go to Step 2.

Step 2: Randomly generate a real number $r_2 \in (0,1)$. If $r_2 < P'_m$, then carry out the non-hub node mutation. Otherwise, keep the chromosome unchanged.

Note that the principle of nearby is not adopted in the mutation operation. This is because the aim of mutation operation is to generate more solutions (chromosomes), which is crucial to search the optimal solution globally. The principle of nearby, however, lowers the diversity of chromosomes.

Selection Process: Each time one chromosome is selected for a new child population, and continuing this process for *pop_size* times, we can get the next population. Let $p_0 = 0$, and $p_i = \sum_{j=1}^{i} a(1-a)^i$, $i = 1, 2, \cdots, pop\_size$, where the parameter $a \in (0,1)$. The selection process is summarized as follows,

Step 1: Set $j = 1$;

Step 2: Randomly generate a number $r \in [0, p_{pop\_size}]$;

Step 3: Find chromosome j satisfying $p_{j-1} \le r < p_j$, and add chromosome j into the chromosome set of the next generation;

Step 4: If $j \ge pop\_size$, stop; otherwise, set $j = j + 1$ and go to Step 2.

The flowchart of the entire heuristic algorithm is shown in Figure 6 in which uncertain data preprocessing is first required to calculate the inverse uncertainty distributions.

4 Numerical Experiments

In this section, we present the computational results of numerical experiments to illustrate the UCCP model and to test the effectiveness of the proposed algorithm. The algorithm is coded in C++ programming language by Microsoft Visual Studio 2010. The work station is a personal computer with Intel(R) Core(TM) i5 3337U 1.80 GHz CPU and 4.00 GB RAM, using the Microsoft Windows 7(64bit) OS.

The experiments are divided into two parts. In the first part, we consider a small-scale problem with 10 nodes, which is referred to as the *basic experiment*. In the basic experiment, we present the detailed solution

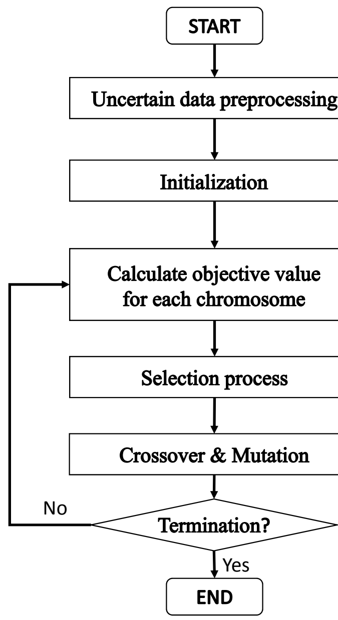


Figure 6: The flowchart of the hybrid intelligent algorithm

process, and compare the efficiency of the ordinary GA and the proposed hybrid intelligent algorithm. In the second part, we further valid the effectiveness and efficiency of the hybrid intelligent algorithm by considering some large-scale problems.

4.1 Basic Experiment

In this section, we consider a p-hub center location problem with 10 nodes in detail. Table 1 lists all the information about travel time on each link. The symbol "$\langle i,j\rangle$" represents the link from node i to node j. In this paper, it is assumed that link "$\langle i,j\rangle$" and link "$\langle j,i\rangle$" are the same. That is to say, the travel time from node i to node j is equal to that from node j to node i, i.e., $\eta_{ij}=\eta_{ji}$, $i,j=1,2,\cdots,10$.

The travel time η_{ij} are all assumed to be zigzag uncertain variables. For example, the data "(11, 14, 17)" indicates that the travel time on link $\langle 1,2\rangle$ is a zigzag uncertain variable $\mathcal{Z}(11,14,17)$. Here 14 is the most possible travel time on link $\langle 1,2\rangle$, 11 is the minimum link travel time and 17 is the maximum link travel time. Note that the travel times on some links may be crisp numbers. For instance, the travel time on link $\langle 2,10\rangle$ is 8. It follows from the definition of uncertain variable that a constant b can be regarded as a special zigzag uncertain variable $\mathcal{Z}(b,b,b)$. Moreover, its inverse uncertainty distribution is equal to $\Phi^{-1}(\beta)\equiv b$ for each $\beta\in(0,1)$.

Suppose that the discount factor $\alpha=0.3$, and the number of hub is 3, i.e., $p=3$. In this work, we set $\beta=0.8$ for the Model UCCP, which means that we want to find the 0.8-optimal p-hub center location. According to Remark 1, the optimal solution of Model UCCP is just that of a deterministic p-hub center location problem. Therefore, we should first calculate the inverse uncertainty distribution $\Phi_{ij}^{-1}(0.8)$ of each travel time η_{ij}, which are all listed in Table 2.

If we use LINGO to solve the deterministic p-hub center location model (12), we can obtain the optimal objective value 28.76 after a computation time of about 200s. Then we use the hybrid intelligent algorithm in Section 3 to solve model (12), which is expected to take less computation time. The parameters in the algorithm are set as follows: the population size of chromosomes $pop\_size=40$, the probability of crossover

Table 1: The data information about travel times on links

link	travel time	link	travel time	link	travel time
$\langle 1,2\rangle$	$(11,14,17)$	$\langle 1,3\rangle$	$(13,16,18)$	$\langle 1,4\rangle$	$(7,9,10)$
$\langle 1,5\rangle$	$(10,13,14)$	$\langle 1,6\rangle$	$(23,26,27)$	$\langle 1,7\rangle$	$(13,15,16)$
$\langle 1,8\rangle$	$(19,23,25)$	$\langle 1,9\rangle$	$(12,15,17)$	$\langle 1,10\rangle$	$(16,19,21)$
$\langle 2,3\rangle$	$(18,19,20)$	$\langle 2,4\rangle$	$(8,11,13)$	$\langle 2,5\rangle$	$(12,15,17)$
$\langle 2,6\rangle$	$(12,14,15)$	$\langle 2,7\rangle$	$(8,10,12)$	$\langle 2,8\rangle$	$(10,12,13)$
$\langle 2,9\rangle$	$(8,11,12)$	$\langle 2,10\rangle$	8	$\langle 3,4\rangle$	$(13,16,18)$
$\langle 3,5\rangle$	$(19,20,21)$	$\langle 3,6\rangle$	$(12,13,14)$	$\langle 3,7\rangle$	$(9,12,13)$
$\langle 3,8\rangle$	$(8,11,13)$	$\langle 3,9\rangle$	$(10,12,13)$	$\langle 3,10\rangle$	$(13,16,18)$
$\langle 4,5\rangle$	$(12,14,15)$	$\langle 4,6\rangle$	$(9,11,12)$	$\langle 4,7\rangle$	$(18,21,23)$
$\langle 4,8\rangle$	$(21,25,27)$	$\langle 4,9\rangle$	$(13,14,16)$	$\langle 4,10\rangle$	$(14,17,18)$
$\langle 5,6\rangle$	$(15,18,20)$	$\langle 5,7\rangle$	$(20,23,24)$	$\langle 5,8\rangle$	$(12,13,14)$
$\langle 5,9\rangle$	$(9,12,13)$	$\langle 5,10\rangle$	$(10,12,13)$	$\langle 6,7\rangle$	$(16,18,19)$
$\langle 6,8\rangle$	$(9,10,11)$	$\langle 6,9\rangle$	$(20,22,23)$	$\langle 6,10\rangle$	$(17,19,21)$
$\langle 7,8\rangle$	$(13,15,17)$	$\langle 7,9\rangle$	9	$\langle 7,10\rangle$	$(12,15,17)$
$\langle 8,9\rangle$	$(19,22,24)$	$\langle 8,10\rangle$	$(12,15,17)$	$\langle 9,10\rangle$	$(8,11,13)$

Table 2: The value of $\Phi_{ij}^{-1}(\beta)$ when $\beta = 0.8$

link	$\Phi_{ij}^{-1}(\beta)$	link	$\Phi_{ij}^{-1}(\beta)$	link	$\Phi_{ij}^{-1}(\beta)$
$\langle 1,2\rangle$	15.8	$\langle 1,3\rangle$	17.2	$\langle 1,4\rangle$	9.6
$\langle 1,5\rangle$	13.6	$\langle 1,6\rangle$	26.6	$\langle 1,7\rangle$	15.6
$\langle 1,8\rangle$	24.2	$\langle 1,9\rangle$	16.2	$\langle 1,10\rangle$	20.2
$\langle 2,3\rangle$	19.6	$\langle 2,4\rangle$	12.2	$\langle 2,5\rangle$	16.2
$\langle 2,6\rangle$	14.6	$\langle 2,7\rangle$	11.2	$\langle 2,8\rangle$	12.6
$\langle 2,9\rangle$	11.6	$\langle 2,10\rangle$	8.0	$\langle 3,4\rangle$	17.2
$\langle 3,5\rangle$	20.6	$\langle 3,6\rangle$	13.6	$\langle 3,7\rangle$	12.6
$\langle 3,8\rangle$	12.2	$\langle 3,9\rangle$	12.6	$\langle 3,10\rangle$	17.2
$\langle 4,5\rangle$	14.6	$\langle 4,6\rangle$	11.6	$\langle 4,7\rangle$	22.2
$\langle 4,8\rangle$	26.2	$\langle 4,9\rangle$	15.2	$\langle 4,10\rangle$	17.6
$\langle 5,6\rangle$	19.2	$\langle 5,7\rangle$	23.6	$\langle 5,8\rangle$	13.6
$\langle 5,9\rangle$	12.6	$\langle 5,10\rangle$	12.6	$\langle 6,7\rangle$	18.6
$\langle 6,8\rangle$	10.6	$\langle 6,9\rangle$	22.6	$\langle 6,10\rangle$	20.2
$\langle 7,8\rangle$	16.2	$\langle 7,9\rangle$	9.0	$\langle 7,10\rangle$	16.2
$\langle 8,9\rangle$	23.2	$\langle 8,10\rangle$	16.2	$\langle 9,10\rangle$	12.2

$P_c = 0.4$, the probability of hub node mutation $P_m = 0.2$, the probability of non-hub node mutation $P'_m = 0.3$, the probability of adopting the principle of nearby $P_{pn} = 0.7$, and the parameter in the rank-based evaluation function $a = 0.05$. A run of the algorithm with 2000 generations shows that the optimal value is 28.76 which corresponds to the p-hub center location $\{2:2,8,10\}, \{4:1,4,6\}$, and $\{9:3,5,7,9\}$. The symbol "$\{2:2,8,10\}$" represents that node 2 is chosen as hub and non-hub node 8 and 10 are assigned to it. The computational time of the algorithm is around $0.15s$, which is much less than that of LINGO. Figure 7 plots the β-optimal p-hub center location for Model UCCP when $\beta = 0.8$.

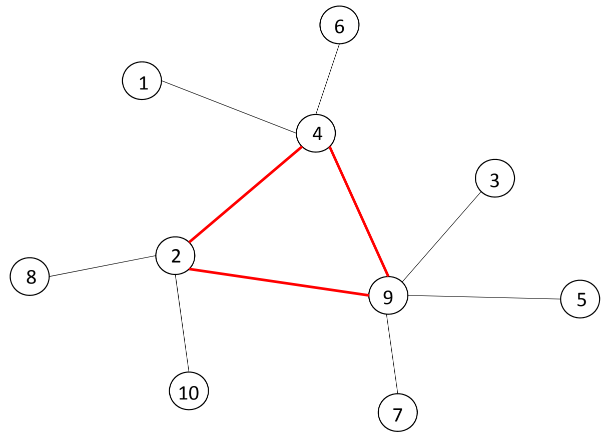


Figure 7: The β-optimal p-hub center location when $\beta = 0.8$

We test the effectiveness of the heuristic algorithm by running it 10 times with the same parameters. The computational results are shown in Table 3 in which the gap is calculated by the formula: (actual value-optimal value)/optimal value×100%. Here actual value represents the obtained objective value of one experiment with given generations, and the optimal value represents the best objective value found by the algorithm. It can be seen that in the 500th generations only 3 experiments do not attain the optimal solution, the gaps of which are all less than 5%; in the 1000th generation, only 1 experiment does not attain the optimal solution, and the gap of which is less than 1%. The results indicate that the hybrid intelligent algorithm designed in Section 3 has good convergence.

Table 3: Computational results of 10 experiments when $\beta = 0.8$

	100 generations		200 gen.		500 gen.		1000 gen.		2000 gen.	
	value	gap	value	gap	value	gap	value	gap	value	gap
1	31.36	9.04%	28.76	0.00%	28.76	0.00%	28.76	0.00%	28.76	0.00%
2	31.78	10.50%	28.96	0.69%	28.76	0.00%	28.76	0.00%	28.76	0.00%
3	31.36	9.04%	30.66	6.60%	28.76	0.00%	28.76	0.00%	28.76	0.00%
4	30.36	5.56%	29.98	4.24%	29.98	4.24%	28.76	0.00%	28.76	0.00%
5	28.76	0.00%	28.76	0.00%	28.76	0.00%	28.76	0.00%	28.76	0.00%
6	30.98	7.71%	28.76	0.00%	28.76	0.00%	28.76	0.00%	28.76	0.00%
7	31.76	10.43%	29.98	4.24%	29.98	4.24%	28.76	0.00%	28.76	0.00%
8	28.96	0.70%	28.96	0.70%	28.96	0.70%	28.96	0.70%	28.76	0.00%
9	29.98	4.24%	29.98	4.24%	28.76	0.00%	28.76	0.00%	28.76	0.00%
10	29.98	4.24%	29.36	2.07%	28.76	0.00%	28.76	0.00%	28.76	0.00%

With different discount factor α, the computational results are shown in Table 4, in which the last column are the optimal objective value of Model UCCP when $\beta = 0.8$. It can be seen that different α will change the hub location, and meanwhile the solutions are relatively robust to α.

Next, we investigate the sensitivity of confidence level β to the optimal objective values. When $\beta = 0.7$, via the proposed algorithm we find that the optimal objective value of Model UCCP is 28.24 with hub

Table 4: Computational results of experiments with different α

α	hub assignment	Opt. obj.
0.05	$\{1:1,4\},\{6:6,8\},\{9:2,3,5,7,9,10\}$	25.20
0.1	$\{1:1,4\},\{6:6,8\},\{9:2,3,5,7,9,10\}$	25.46
0.2	$\{3:3,8\},\{4:1,4,6\},\{9:2,5,7,9,10\}$	27.32
0.3	$\{2:2,8,10\},\{4:1,4,6\},\{9:3,5,7,9\}$	28.76
0.4	$\{2:2,8,10\},\{4:1,4,6\},\{9:3,5,7,9\}$	30.28

location $\{3:3,8\}$, $\{4:1,4,6\}$ and $\{9:2,5,7,9,10\}$. Figure 8 illustrates the β-optimal p-hub center location when $\beta = 0.7$, which is different from that when $\beta = 0.8$.

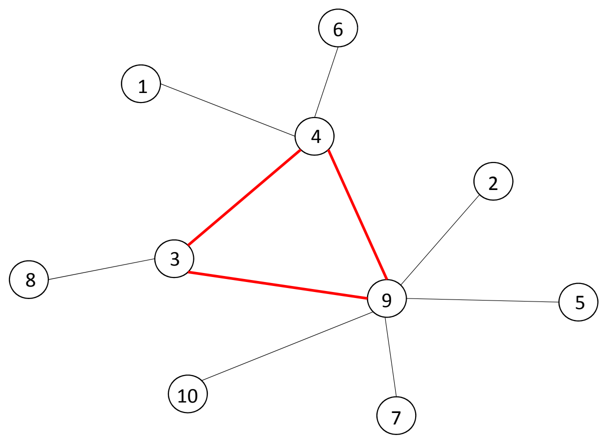


Figure 8: The β-optimal p-hub center location when $\beta = 0.7$

The experiments are conducted by solving Model UCCP for different confidence levels β and the computational results are summarized in Table 5 and in Figure 9. From the figure, we can observe that the optimal objective values zigzag increase with the increase of β, the kink point of which is at $\beta = 0.5$.

Table 5: The optimal objective values obtained by UCCP

Confidence level β	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90	0.95	0.99
β-cost	22.96	24.02	25.08	26.14	27.20	27.72	28.24	28.76	29.28	29.54	29.75

Finally, we discuss the introduction of the principle of nearby. Consider the deterministic p-hub center location problem with link travel times listed in Table 2. As has been stated, the optimal solution of this problem is $\{2:2,8,10\}$, $\{4:1,4,6\}$, and $\{9:3,5,7,9\}$. Choosing nodes 2, 4 and 9 as hub nodes, Table 6 presents the link travel times between a hub node and a non-hub node. The symbol "H.k" represents "hub node k", and the symbol "N.i" represents "non-hub node i". In each column, the minimum link travel time is marked in red. Note that in the optimal solution, the non-hub nodes are all assigned to the hub node that is nearest to them. Hence, in this example the principle of nearby produces the optimal solution.

The influence of P_{pn} is also investigated in our work. Keeping the other parameters unchanged, *i.e.*, $pop\_size = 40$, $P_c = 0.4$, $P_m = 0.2$, $P'_m = 0.3$, $a = 0.05$ and $p = 3$, we repeat the algorithm 30 times

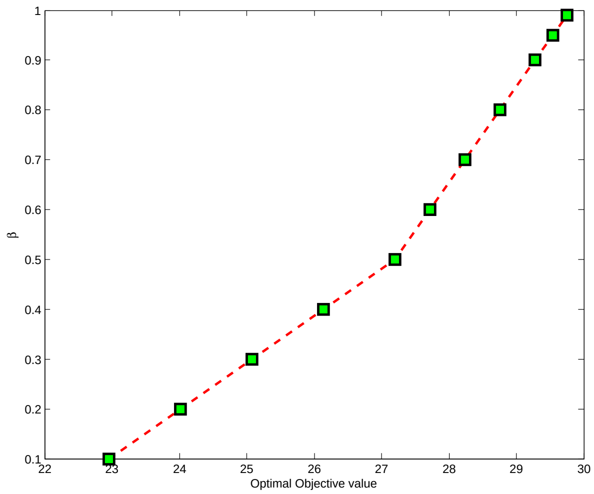


Figure 9: The sensitivity of confidence level to the optimal objective values

Table 6: Link travel time between a non hub node and a hub node

	N.1	N.3	N.5	N.6	N.7	N.8	N.10
H.2	15.8	19.6	16.2	14.6	11.2	12.6	8.0
H.4	9.6	17.2	14.6	11.6	22.2	26.2	17.6
H.9	16.2	12.6	12.6	22.6	9.0	23.2	12.2

for different P_{pn}. Table 7 presents the numbers of the experiments that attain the optimal solution within given number of generations. For example, when $P_{pn} = 0.1$, there are 8 experiments that attain the optimal solution within 1000 generations, and 11 experiments within 2000 generations. It can be seen that the result is becoming better as P_{pn} increases to 0.7. For example, the quality of the result is nearly the same when $P_{pn} = 0.7$ and $P_{pn} = 0.9$. It needs to be emphasized that when $P_{pn} = 0$, the hybrid intelligent algorithm degenerates to an ordinary genetic algorithm. The first row of Table 7 shows that for the ordinary genetic algorithm, there is no experiment attaining the optimal solution within 2000 generations. This again highlights the effectiveness and efficiency of the proposed solution procedure.

Table 7: Results of 30 experiments for different P_{pn}

P_{pn}	100 generation	200 gen.	500 gen.	1000 gen.	2000 gen.
0.0	0	0	0	0	0
0.1	2	5	7	8	11
0.3	2	4	11	15	20
0.5	0	2	14	20	25
0.7	3	10	17	25	30
0.9	5	11	14	24	30

4.2 Large-scale Experiment

As shown in the previous section, the computation time is quite short when the scale of the problem is small, *i.e.*, $n = 10$. In this section, we carry out experiments with different n, increasing from 20 to 100, to verify the effectiveness and efficiency of the proposed hybrid intelligent algorithm. The probability parameters in the algorithm are set as follows: the probability of crossover $P_c = 0.4$, the probability of hub node mutation $P_m = 0.2$, the probability of non-hub node mutation $P'_m = 0.3$, the probability of adopting the principle of nearby $P_{pn} = 0.7$. For different n and p, different *pop_size* of chromosomes and generation number (*gen_num*) are used, which are listed in the third and the fourth column of Table 8 respectively.

For each pair of (n, p), we carry out 10 experiments. In each experiment, the travel times between any two nodes are generated randomly from the uniform distribution $U[5, 100]$. For each experiment, we iterate 200000 times, and the obtained objective value is used to calculate the gap. The average gap and the average computation time of the 10 experiments are then calculated, which are listed in the fifth and the sixth column of Table 8 respectively.

Table 8: Computational results with different n and p

n	p	*pop_size*	*gen_num*	*avg_gap*	CPU[s]
	3	40	6000	0.00%	2.1
20	4	60	8000	0.20%	2.5
	5	80	10000	0.18%	3.7
	3	40	6000	0.00%	3.4
40	4	60	8000	0.34%	4.9
	5	80	10000	0.41%	6.3
	4	60	8000	0.20%	9.6
60	5	80	10000	0.52%	12.2
	6	100	12000	0.60%	22.0
	5	80	12000	1.56%	49.3
80	6	100	14000	1.90%	64.5
	7	120	16000	3.80%	103
	6	80	16000	2.30%	122
100	7	120	18000	4.07%	175
	8	160	20000	5.60%	240

Obviously, as n increase, both the average gaps and the average computation times increase. When n is medium, *i.e.*, $n \leq 60$, the average gaps are all less than 0.6%, and the computation times are all less than $25s$, which indicates that the hybrid intelligent algorithm converges to the optimal solution fast. When n is large, *i.e.*, $n \geq 80$, the convergency becomes weak, *i.e.*, the average gap may be greater than 5% after a computation time of $240s$. This indicates that the hybrid intelligent algorithm is more efficient for the problems of medium-scale.

5 Conclusions

This paper extended the classical p-hub center location problem to the uncertain setting by describing the travel times as uncertain variables. We first derived the uncertainty distribution of the objective function in the presence of uncertainty and then applied chance constrained programming approach to formulate a new uncertain programming model. The deterministic equivalent models were obtained based on the given uncertainty distribution of each travel time. Due to the computational complexity of the problem, we designed a hybrid intelligent algorithm by combining principle of nearby into a genetic algorithm to solve the proposed models. The effectiveness and efficiency of the proposed heuristic solution procedure was verified by some numerical experiments. This modelling methodology and the algorithm may also be applied to other optimization problems.

Appendix

This appendix introduces the basic definitions and results on uncertain measure and uncertain variable. Uncertain measure was proposed by Liu (2007) to describe a belief degree that a possible event happens. Let $\mathcal{L}$ be a σ-algebra on a nonempty set Γ. A set function $\mathcal{M}\colon \mathcal{L} \to [0,1]$ is called an uncertain measure if it satisfies: (1) $\mathcal{M}\{\Gamma\} = 1$ for the universal set Γ; (2) $\mathcal{M}\{\Lambda\} + \mathcal{M}\{\Lambda^c\} = 1$ for any event Λ; (3)For every countable sequence of events $\Lambda_1, \Lambda_2, \cdots$, we have

$$\mathcal{M}\left\{\bigcup_{i=1}^{\infty} \Lambda_i\right\} \le \sum_{i=1}^{\infty} \mathcal{M}\left\{\Lambda_i\right\}.$$

The triple $(\Theta, \mathcal{P}, \mathcal{M})$ is called an uncertainty space.

Let $(\Gamma_k, \mathcal{L}_k, \mathcal{M}_k)$ be uncertainty spaces for $k = 1, 2, \cdots$. Then the product uncertain measure $\mathcal{M}$ is provided by Liu (2010) as an uncertain measure satisfying

$$\mathcal{M}\left\{\prod_{k=1}^{\infty} \Lambda_k\right\} = \bigwedge_{k=1}^{\infty} \mathcal{M}_k\{\Lambda_k\}$$

where Λ_k are arbitrarily chosen events from $\mathcal{L}_k$ for $k = 1, 2, \cdots$, respectively.

An uncertain variable was proposed by Liu (2007) to model a quantity under imprecision, which is a measurable function from an uncertainty space $(\Gamma, \mathcal{L}, \mathcal{M})$ to the set of real numbers. That is, if ξ is an uncertain variable, then for any Borel set B of real numbers, the set $\{\xi \in B\} = \{\gamma \in \Gamma \mid \xi(\gamma) \in B\}$ is an event. For any $x \in \Re$, $\Phi(x) = \mathcal{M}\{\xi \le x\}$ is called the uncertainty distribution of ξ.

Definition 2. *(Liu, 2010) An uncertainty distribution $\Phi(x)$ is said to be regular if it is a continuous and strictly increasing function with respect to x at which $0 < \Phi(x) < 1$, and*

$$\lim_{x \to -\infty} \Phi(x) = 0, \quad \lim_{x \to +\infty} \Phi(x) = 1.$$

The inverse function $\Phi^{-1}(\alpha)$ is called the inverse uncertainty distribution of ξ if it exists and is unique for each $\alpha \in (0,1)$.

For example, linear uncertainty distribution, zigzag uncertainty distribution, normal uncertainty distribution, and lognormal uncertainty distribution are all regular. Inverse uncertainty distribution also plays a crucial role in operations of independent uncertain variables.

Definition 3. *(Liu, 2010) The uncertain variables $\xi_1, \xi_2, \cdots, \xi_n$ are said to be independent if*

$$\mathcal{M}\left\{\bigcap_{i=1}^{n}(\xi_i \in B_i)\right\} = \bigwedge_{i=1}^{n} \mathcal{M}\{\xi_i \in B_i\}$$

for any Borel set $B_1, B_2, \cdots, B_n$ of real numbers.

Lemma 1. *(Liu, 2007; operational law) Let $\xi_1, \xi_2, \cdots, \xi_n$ be independent uncertain variables with regular uncertainty distributions $\Phi_1, \Phi_2, \cdots, \Phi_n$, respectively. If $f(x_1, x_2, \cdots, x_n)$ is strictly increasing with respect to $x_1, x_2, \cdots, x_n$, then $\xi = f(\xi_1, \xi_2, \cdots, \xi_n)$ is an uncertain variable with an uncertainty distribution*

$$\Psi(x) = \sup_{f(x_1, x_2, \cdots, x_n) = x} \min_{1 \leq i \leq n} \Phi_i(x_i),$$

and with an inverse uncertainty distribution

$$\Psi^{-1}(\alpha) = f(\Phi_1^{-1}(\alpha), \Phi_2^{-1}(\alpha), \cdots, \Phi_n^{-1}(\alpha)).$$

It is worth pointing out that the product axiom is different from that of probability measure. Therefore, the independence has a different definition in the framework of uncertainty theory. By the product axiom and the definition of independence, the operation law of uncertain variables is absolutely different from that of random variables.

Acknowledgements

The first author was supported by the National Natural Science Foundation of China (Nos.71401008, 71271022, and U1404701), the Fundamental Research Funds for the Central Universities (No.2014RC038), the Soft Science Research Program of Henan Province (No.152400410447), and State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University (No.RCS2015ZZ003). The second author was supported in part by National Natural Science Foundation of China (Nos. 71371019 and 71332003), and in part by the Fundamental Research Funds for the Central Universities (No. YWF-16-BJY-20).

References

[1] Bashiri, M., Mirzaei, M., & Randall, M. (2013). Modeling fuzzy capacitated p-hub center problem and a genetic algorithm solution, *Applied Mathematical Modelling*, 37(5), 3513-3525.

[2] Campbell, J.F. (1994). Integer programming formulations of discrete hub location problems, *European Journal of Operational Research* 72, 387-405.

[3] Campbell, A.M., Lowe, T.J., & Zhang, L. (2007). The p-hub center allocation problem, *European Journal of Operational Research*, 176(2), 819-835.

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

[5] Charnes A., & Cooper W.W. (1959). Constrained-chance programming, *Management Science*, 6(1), 73-79.

[6] Chen X., Kar S., & Ralescu D. 2012. Cross-entropy measure of uncertain variables, *Information Sciences*, 201, 53-60.

[7] Farahani, E.Z., Hekmatfar, M., Arabani, A.B., & Nikbakhsh, E. (2013). Hub location problems: A review of models, classification, solution techniques, and applications, *Computers & Industrial Engineering*, 64, 1096-1109.

[8] Gao, Y. (2011). Shortest path problem with uncertain arc lengths. *Computers and Mathematics with Applications* 62(6), 2591-2600.

[9] Gao, Y. (2012). Uncertain models for single facility location problems on networks, *Applied Mathematical Modelling*, 36(6), 2592-2599.

[10] Gao, Y., Yang L., Li S., & Kar, S. (2015). On distribution function of the diameter in uncertain graph, *Information Sciences*, 296, 61-74.

[11] Gen, M., & Cheng, R. (2000). *Genetic Algorithms and Engineering Optimization*, John Wiley & Sons.

[12] Han, S., Peng, Z., & Wang, S. (2014). The maximum flow problem of uncertain network, *Information Sciences*, 265(1), 167-175.

[13] Holland, J. (1975). *Adaptatin in Natural and Artificial System*, University of Michigan Press, Ann Arbor.

[14] Hult, E., Jiang, H., & Ralph, D. (2014). Exact computational approaches to a stochastic uncapacitated single allocation p-hub center problem, *Computational Optimization and Applications*, 59(1), 185-200.

[15] Liu B., *Uncertainty Theory*, 2nd ed., Springer-Verlag, Berlin, 2007.

[16] Liu B., *Uncertainty Theory: A Branch of Mathematics for Modeling Human Uncertainty*, Springer-Verlag, Berlin, 2010.

[17] Sim, T., Lowe, T.J., & Thomas, B.W. (2009). The stochastic p-hub center problem with service-level constraints, *Computers and Operations Research*, 36, 3166-3177.

[18] Topcuoglu, H., Corut, F., Ermis, M., &Yilmaz, G. (2005). Solving the uncapacitated hub location problem using genetic algorithms, *Computers & Operations Research*, 32, 967-984.

[19] Wen, M., Qin, Z., & Kang, R. (2014). The α-cost minimization model for capacitated facility location-allocation problem with uncertain demands. *Fuzzy Optimization and Decision Making*, doi:10.1007/s10700-014-9179-z.

[20] Yang, K., Liu, Y., & Zhang, X. (2011). Stochastic p-hub center problem with discrete time distributions, *Lecture Notes in Computer Science*, 6676(2), 182-191.

[21] Yang, K., Liu, Y., & Yang, G. (2013a). An improved hybrid particle swarm optimization algorithm for fuzzy p-hub center problem, *Computers & Industrial Engineering*, 64, 133-142.

[22] Yang, K., Liu, Y., & Yang, G. (2013b). Solving fuzzy p-hub center problem by genetic algorithm incorporating local search, *Applied Soft Computing*, 13, 2624-2632.

[23] Yang, K., Liu, Y., & Yang, G. (2014). Optimizing fuzzy p-hub center problem with generalized value-at-risk criterion, *Applied Mathematical Modelling*, doi:10.1016/j.apm.2014.01.009.

Highlights

A chance constrained programming approach for uncertain p-hub center location problem

- We first investigate the p-hub center problem with subjective imprecision.
- We formulate a chance constrained programming model for uncertain p-hub center location problem.
- We give the analytical forms of the proposed model base on uncertainty theory.
- We present principle of nearby and design a hybrid intelligent algorithm to solve the uncertain p-hub center models.