Computers & Operations Research 62 (2015) 224–236

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

A genetic algorithm for the uncapacitated single allocation planar hub location problem

Haluk Damgacioglu *, Derya Dinler, Nur Evin Ozdemirel, Cem Iyigun

Industrial Engineering Department, Middle East Technical University Dumlupinar Bulvari, 06800 Ankara, Turkey

ARTICLE INFO

Available online 18 September 2014

Keywords:
Genetic algorithms
Hub location problems
Planar hub location problems

ABSTRACT

Given a set of n interacting points in a network, the hub location problem determines location of the hubs (transfer points) and assigns spokes (origin and destination points) to hubs so as to minimize the total transportation cost. In this study, we deal with the uncapacitated single allocation planar hub location problem (PHLP). In this problem, all flow between pairs of spokes goes through hubs, capacities of hubs are infinite, they can be located anywhere on the plane and are fully connected, and each spoke must be assigned to only one hub. We propose a mathematical formulation and a genetic algorithm (PHLGA) to solve PHLP in reasonable time. We test PHLGA on simulated and real life data sets. We compare our results with optimal solution and analyze results for special cases of PHLP for which the solution behavior can be predicted. Moreover, PHLGA results for the AP and CAB data set are compared with other heuristics.

1. Introduction

Hub location problems (HLPs) are defined on a network such that there are n interacting points called spokes and p centers of transportation named hubs. The links in a hub location network are of three types. These are: (1) collection links from spokes (origin points) to hubs, (2) inter-hub transfer links between pairs of hubs, and (3) distribution links from hubs to spokes (destination points). The hub location network is first defined in 1969 by Goldman [1]. An example hub location network is shown in Fig. 1 where nodes i and j represent spokes, and k and l are hubs. HLP determines the hub locations and assigns spokes to hubs so as to minimize flow and distance weighted transportation cost in the network.

In a hub location network, the supply from some of the origin points is collected at a hub, transferred to another hub together, and then distributed to the demand points. The main motivation of using a hub-spoke network is to take advantage of the cost reductions between hubs due to the economies of scale. In most cases, the aggregated flow between hubs reduces the total transportation cost compared to direct shipment between all pairs of spokes. Hub-spoke networks are used in various industries such as airlines, shipment, cargo delivery, and telecommunication. For example, since larger trucks are used between hubs, the unit transportation cost decreases in shipping goods. In airline industry and telecommunication networks, use of hubs eliminates the need for all pairwise connections between spokes, reducing the operational costs significantly.

Since hub-spoke networks have many practical uses in various industries, HLP is widely studied by researchers and there are many variants of the problem. For example, each spoke can be assigned to a single hub or multiple hubs, hubs can be uncapacitated or have limited capacity. One variant imposes a constraint on the hub locations such that hubs can only be located on predetermined points, typically some of the origin and destination points. This variant is called the discrete hub location problem (DHLP). If hub locations are not restricted and they can be located anywhere on the plane, the problem is named as the planar hub location problem (PHLP).

Most of the studies in this area focus on solving the discrete version of the problem. DHLP is known to be NP-hard [1]. Therefore, researchers developed some heuristics to find a good solution in reasonable time. DHLP was initially studied by O'Kelly [2]. He proposed the first mathematical programming formulation of the problem and developed two heuristics, namely HEUR-1 and HEUR-2. Klincewicz first developed an exchange heuristic in 1991 [3], and then a Tabu Search and a GRASP heuristic [4]. Campbell [5] introduced two heuristics for the single allocation DHLP. The solutions found by these heuristics are obtained by modifying the solutions for the multiple allocation version of the problem. Moreover, Ernst and Krishnamoorthy [6], and Abdinnour-Helm [7] proposed simulated annealing metaheuristics to solve the problem. In 1998, Ernst and Krishnamoorthy [8] developed a branch

* Corresponding author.
E-mail addresses: hdamgacioglu@gmail.com (H. Damgacioglu), dinler@metu.edu.tr (D. Dinler), nurevin@metu.edu.tr (N. Evin Ozdemirel), iyigun@metu.edu.tr (C. Iyigun).

http://dx.doi.org/10.1016/j.cor.2014.09.003
0305-0548/

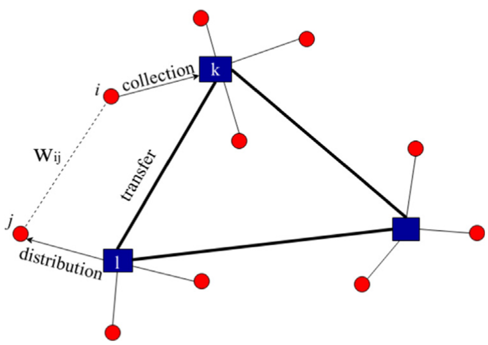

Fig. 1. An example of hub location networks.

and bound algorithm based on the shortest path problem. Topcuoglu et al. [9] and Kratica et al. [10] approached the problem using Genetic Algorithms. Ilic et al. [11] proposed a variable neighborhood search heuristic to solve the problem, which solves instances with up to 1000 nodes in reasonable time.

The planar version of the hub location problem is underappreciated in the literature compared to the discrete version. There are only a few studies on the PHLP. O'Kelly [12] approached the planar hub location based on clustering. He solved instances with up to 500 nodes and 9 hubs. Campbell [13] proposed three strategies to locate the terminals by continuous approximation of freight carrier with increasing demand problem. Aykin and Brown [14] developed a modified location–allocation algorithm to solve the facility location problem in which facilities interact with each other.

Alev and Kara [1], and Campbell and O'Kelly [15] provide detailed reviews of HLP studies.

Our motivation for studying PHLP is threefold. First of all, PHLP is mathematically challenging because it has a non-differentiable objective function, which is not easy to optimize. Secondly, it can be useful in some real world problems where locations can be determined in continuous space, such as city logistics, telecommunications, and cargo delivery systems. For example, a shipping or cargo delivery center within a city can be located at almost anywhere. The same is true for GSM base stations or wireless routers to be located in rural areas. Thirdly, PHLP is more sensitive to problem parameters (amount and cost of flow) than DHLP, and PHLP solutions may provide valuable insight for choosing hub locations in DHLP.

In this study, we give a mathematical formulation of the uncapacitated single allocation PHLP and propose a genetic algorithm to solve it. In this version of the PHLP, all flow between pairs of spokes go through the hubs, capacities of the hubs are infinite, they are fully connected, and each spoke must be assigned to only one hub. The problem is reduced to the multifacility location problem (MFLP) if the costs of inter-hub transfer links are zero, and the costs of collection and distribution links are equal. Megiddo and Supowit [16] have proved that MFLP is NP-Hard. Thus, PHLP is also NP-Hard and a mathematically challenging problem requiring efficient heuristic solution procedures.

To test how our algorithm behaves, we use some simulated data sets representing special cases of PHLP for which the optimal solution can be found. The solution quality of the algorithm is promising and it solves large problem instances in reasonable time. Then, we work on some real world data sets from the literature and compare our results with other heuristics. To the best of our knowledge, we present the first results for the planar versions of the CAB and AP [2,6] data sets. Finally, to show how PHLP solutions can provide insight for DHLP, we compare solutions obtained by modifying the PHLP solutions with the optimal solutions for medium sized DHLP instances. We also present modified PHLP solutions for large DHLP instances.

The rest of the paper is organized as follows. In Section 2, we propose a mathematical programming formulation for the PHLP. Section 3 describes the details of our genetic algorithm. Computational results for simulated and real world data sets are presented in Section 4. Section 5 includes concluding remarks for the study.

2. Mathematical formulation of PHLP

Consider a hub-spoke network consisting of interacting points and three types of links for collection, transfer, and distribution. We assume the following for PHLP.

- The number of hubs is given.
- Hubs are fully connected, and all flow between pairs of spokes go through hubs, i.e. direct shipment between spokes is not allowed.
- Hubs have infinite capacity.
- Each spoke must be assigned to only one hub.

The mathematical programming formulations of PHLP and DHLP are closely related. Therefore, the formulation of DHLP should be primarily investigated to understand and formulate PHLP. We first give the notation commonly used in the formulation of both DHLP and PHLP.

n	The number of nodes (spokes) in the hub-spoke network.
p	The number of hubs.
i,j	Indices for nodes, $i,j=1,\ldots,n$.
k,l	Indices for nodes, $k,l=1,\ \ldots,\ p$.
w_{ij}	Weight representing the amount of flow from node i to node j.
α_c	Collection cost (per unit flow and unit distance) from an origin to a hub.
α_d	Distribution cost (per unit flow and unit distance) from a hub to a destination.
α_t	Transfer cost (per unit flow and unit distance) between a pair of hubs such that $\alpha_t \le \min\{\alpha_c, \alpha_d\}$.
d_{ik}	Distance from node i to node k.

The first mathematical programming formulation of the DHLP was proposed by O'Kelly [2]. In his formulation x_{ik} represents assignment of spokes to hubs for each $i \neq k$, that is x_{ik} takes the value one if node i is assigned to hub k, zero otherwise. When $i=k$, $x_{kk}=1$ means that the node is selected as hub. Then, O'Kelly's [2] mathematical programming formulation of DHLP is as follows.

Problem 1.

$$\min \quad \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\left\{\sum_{k=1}^{n}\alpha_c d_{ik}x_{ik} + \sum_{k=1}^{n}\alpha_d d_{jl}x_{jl} + \sum_{k=1}^{n}\sum_{l=1}^{n}\alpha_t d_{kl}x_{ik}x_{jl}\right\} \tag{1}$$

$$\text{subject to} \quad (n-p+1)x_{kk} - \sum_{i=1}^{n} x_{ik} \ge 0 \quad \forall k \tag{2}$$

$$\sum_{k=1}^{p} x_{ik} = 1 \quad \forall i \tag{3}$$

$$\sum_{k=1}^{n} x_{kk} = p \tag{4}$$

$$x_{ik} \in \{0,1\} \quad \forall i,k \tag{5}$$

Objective function (1) is a function of the hub selection and assignment decisions. The three terms represent collection costs from spokes to hubs, distribution costs from hubs to spokes, and transfer costs between hubs. Constraint set (2) is used to assign non-hub nodes to hub nodes. Note that a hub node cannot be assigned to another hub, but there can be a hub with no spokes. Set (3) ensures that each spoke is assigned to exactly one hub. Constraint set (4) limits the number of hubs to exactly p.

The assignment of spokes to hubs is similar for DHLP and PHLP. However, there is a crucial difference between the two versions. While hubs can be located only on given nodes in DHLP, they can be located anywhere on the two (or higher) dimensional space in PHLP. Therefore, there is a continuous decision (determining hub locations) in addition to a discrete decision (assigning spokes to hubs) in PHLP. We define our decision variables accordingly. For the assignment part, a binary decision variable is used as in the discrete version of the problem: x_{ik} takes the value one if node i is assigned to hub k, zero otherwise. For determining hub locations, a location vector is needed. Since we work on the plane, the vector has two elements, but this can be generalized to higher dimensions. Let $\mathbf{y}_k = (y_k^1, y_k^2)$ represent the coordinates of hub k.

Generally speaking, Euclidean distance is a well-known distance measure in the continuous facility location problems. To the best of our knowledge, it is the only measure used in PHLPs, although any other norm distance can also be used. Given the location of node i by the vector $\mathbf{a}_i = (a_i^1, a_i^2)$ to measure distance between hub k and node i we used Euclidean distance calculated as follows.

$$d(\mathbf{y}_k, \mathbf{a}_i) = \|\mathbf{y}_k - \mathbf{a}_i\| = \sqrt{(y_k^1 - a_i^1)^2 + (y_k^2 - a_i^2)^2} \tag{6}$$

Then, the mathematical programming formulation of PHLP is as follows.

Problem 2.

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\left\{\sum_{k=1}^{p} \alpha_c d(\mathbf{y}_k, \mathbf{a}_i)x_{ik} + \sum_{l=1}^{p} \alpha_d d(\mathbf{y}_l, \mathbf{a}_j)x_{jl} + \sum_{k=1}^{p}\sum_{l=1}^{p} \alpha_t d(\mathbf{y}_k, \mathbf{y}_l)x_{ik}x_{jl}\right\} \tag{7}$$

$$\text{subject to} \quad \sum_{k=1}^{p} x_{ik} = 1 \quad \forall i \tag{8}$$

$$\sum_{i=1}^{n} x_{ik} \geq 1 \quad \forall k \tag{9}$$

$$x_{ik} \in \{0,1\} \quad \forall i,k \tag{10}$$

We use this formulation to succinctly define the problem and introduce the notation. Objective function (7) minimizes the total transportation cost as the sum of the collection, distribution, and transfer costs. Constraint set (8) ensures that each node is assigned to only one hub. Set (9) states that there should be at least one spoke assigned to each hub so that there are exactly p hubs used. Constraint (10) is integrality constraint for the decision variables. Note that there is no need in P2 for the constraint set (2) used in P1.

Aykin and Brown [14] also give a formulation for PHLP similar to P2. Their formulation is based on mathematical programming formulation of the location–allocation problem. In this formulation, they assume that the unit collection and distribution costs are equal. This allows them to define a discount factor for the unit transfer cost between hubs. That is, $\alpha_t = \tau\alpha_c = \tau\alpha_d$ where τ is the discount factor such that $\tau \in [0,1]$. They also assume that the weights are symmetrical such that $w_{ij} = w_{ji}$ for all i,j. In our formulation, we use different cost parameters for collection, distribution and transfer links and we assume that weights are not necessarily symmetrical. Therefore, our formulation is more general and reflects real life problems better than Aykin and Brown's formulation. Moreover, they do not use the constraint set (9) in P2. This means that there can be some hubs without any spokes. It should be noted that increasing p decreases cost due to discount when a distance measure satisfying the triangular inequality is used. In such a case, fixing p reduces the solution space. Therefore, it can be concluded that P2 is more effective than Aykin and Brown's formulation. If the distance measure used does not satisfy the triangular inequality, reduction in cost is not guaranteed. In this sense, their formulation may find a better solution with fewer hubs for the same p value.

3. The PHLGA algorithm

Genetic Algorithms (GAs) are population based metaheuristics widely used to solve combinatorial problems. In the last decades, GAs are used to solve many problems and found to be effective and robust search methods. In the literature there are some studies that propose GAs to solve the DHLP [9,10].

Since main idea of GAs is based on natural selection process, the terms used in these algorithms are borrowed from this process. In GAs, an individual or a chromosome represents a solution of the problem. A chromosome is encoded as a string of genes, which represent solution components. Although a solution is traditionally encoded by a binary string, higher cardinality domains can also be used.

A genetic algorithm starts with an initial population of solutions, and this population evolves through generations. In each generation, individuals with higher fitness are selected as parents to reproduce using a crossover operator. Offspring generated go through mutation and replace all or inferior parents. This way, good features are allowed to spread over the population, mixing with other good features and eventually producing superfit individuals.

As the main concepts of GAs are applied to a specific problem, certain choices should be made concerning the encoding or solution representation scheme, fitness function, initial population generation, parent selection, reproduction through crossover and mutation operators, and parent replacement. The details of our algorithm for PHLP are given below.

3.1. Representation scheme

In PHLGA, two sets of decision variables (assignment and location variables) are defined in different domains. Therefore, we use different representation schemes for each set of decision variables.

- We represent the assignment of spokes to hubs with a $1 \times n$ vector. In this representation, ith element of the vector shows the hub number to which node i is assigned. Therefore, each gene can take a value from the set $\{1, \ldots, p\}$. An example of the assignment scheme for a problem with $n=7$ nodes and $p=3$ hubs is

1	2	3	2	2	1	3

- We represent the hub locations by using a $p \times 2$ matrix. kth row of the matrix shows the coordinates of hub k. An example of hub location representation for a problem with $p=3$ hubs is

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \end{bmatrix} = \begin{bmatrix} 8 & 12 \\ 4 & 1 \\ 20 & 4 \end{bmatrix}$$

3.2. Fitness function

We use the objective function (7) of problem P2 as the fitness function. Since we want to minimize the cost, lower values of the objective function imply higher fitness.

3.3. Initial population generation

Traditionally the initial population is generated randomly. In PHLGA, we generate *pop_size* individuals first by locating the hubs on the plane randomly and then by assigning each node to the closest hub in the procedure of generating the initial population. Random hub locations are uniformly distributed in the minimum rectangular area encapsulating the spoke locations.

3.4. Parent selection

In any generation of PHLGA, all individuals in the population are allowed to reproduce. For mating two parents, we first shuffle the individuals in the population, and then select parents from this random sequence one after another. That is, individuals 1 and 2 mate, so do 2 and 3, 3 and 4, and so on. This parent selection method allows each individual in the population have a chance to reproduce offspring twice.

3.5. Crossover operators

The crossover operator is used to produce offspring by recombining genes from selected parents. We use two different crossover operators for discrete assignment and continuous location decisions. The assignment operator tries to preserve common genes (hub numbers) in parents. However, hubs are numbered arbitrarily and seemingly different chromosomes may actually represent the same solution. What is important is not the actual hub numbers, but assignment of a certain set of nodes to the same hub. Therefore, we resort to renumbering of hubs before applying the assignment crossover [17]. Without renumbering, crossover could not pass good solution components (building blocks) from parents to offspring. Offspring generated would essentially be random and the GA would approach random search.

In the renumbering procedure, a $p \times p$ distance matrix $\mathbf{D}$ is calculated from hub locations of two parents. Element d_{kl} of $\mathbf{D}$ is the distance between hub k from parent A and hub l from parent B. Then, taking parent A as the frame of reference, hubs of parent B are renumbered to obtain parent B'. For hub k in parent A, the closest hub in parent B is found as $l' = \arg\min_l \{d_{kl}\}$ and hub l' in parent B is renumbered as k. Renumbering is also reflected in the assignment vector by updating the hub numbers in B to new numbers in B'. Hence, B and B' represent identical solutions in general.

We do not know which of the original parent has a better fitness, therefore we carry out renumbering starting with the other parent as well. Hence, hubs of parent A are also renumbered by taking parent B as the frame of reference, that is $k' = \arg\min_k \{d_{kl}\}$ is used to renumber hub k' in A as hub l in A'. An example of renumbering for a problem where $n=7$ and $p=3$ is given in Fig. 2. Note that parent A' does not represent the same solution as parent A because $k'=3$ for both hubs $l=1$ and $l=2$ of parent B. In this case hub 3 of A becomes both hub 1 and hub 2 in A'. Renumbering may result in fewer than p hubs as in this example, but we make sure that the final PHLGA solution has exactly p hubs each with at least one spoke.

After parents A' and B' are obtained by renumbering hubs in parents A and B, first the assignment and then the location operators are applied. Offspring 1 is generated from parents A and B', offspring 2 from parents A' and B. The assignment crossover works as follows. If a spoke is assigned to the same hub in both parents, it is directly assigned to that hub in the offspring. If the hub numbers of a spoke are different in two parents, then one of them is selected at random with equal probability. If there are fewer than p hubs in the resulting offspring, we repeat random selection of hubs from parents until there are exactly p hubs. An example of the assignment crossover is given in Fig. 3.

In the location crossover, hub locations of offspring are determined by taking the affine combination of hub locations of parents with a parameter μ as in the following equation:

$$\begin{aligned} \mathbf{Y}_{offspring1} &= \mu \mathbf{Y}_A + (1-\mu)\mathbf{Y}_{B'} \\ \mathbf{Y}_{offspring2} &= \mu \mathbf{Y}_B + (1-\mu)\mathbf{Y}_{A'} \end{aligned} \tag{11}$$

μ is randomly generated from normal distribution whose mean and variance are 0.5. Using the parameter μ helps search the solution space better for hub locations. For example, location of hub k in offspring 1 will be on the line segment connecting $\mathbf{y}_{k,A}$ and $\mathbf{y}_{k,B'}$ with probability 0.68. With probability 0.32 it will be outside the line segment. However, if with reference to μ value, any hub is located outside the rectangular region defined by the spokes, we regenerate μ and hub locations until they fall inside.

3.6. Mutation operators

The mutation operator is used to search unexplored solution space by changing solution components with a small probability. As in crossover, we use different mutation operators for assignment and location. In the assignment operator, spoke i is reassigned to the least cost hub k_i^* by the following equation:

$$k_i^* = \arg\min_k \left\{ collection_{ik} + distribution_{ki} + \sum_l transfer_{kl} \right\} \tag{12}$$

$$\text{where } collection_{ik} = \sum_{j=1}^{n} w_{ij} \alpha_c d(\mathbf{y}_k, \mathbf{a}_i) \tag{13}$$

$$distribution_{ki} = \sum_{j=1}^{n} w_{ji} \alpha_d d(\mathbf{y}_k, \mathbf{a}_i) \tag{14}$$

$$\sum_l transfer_{kl} = \sum_{j=1}^{n} w_{ij} \sum_{l=1}^{p} \alpha_t d(\mathbf{y}_k, \mathbf{y}_l) \tag{15}$$

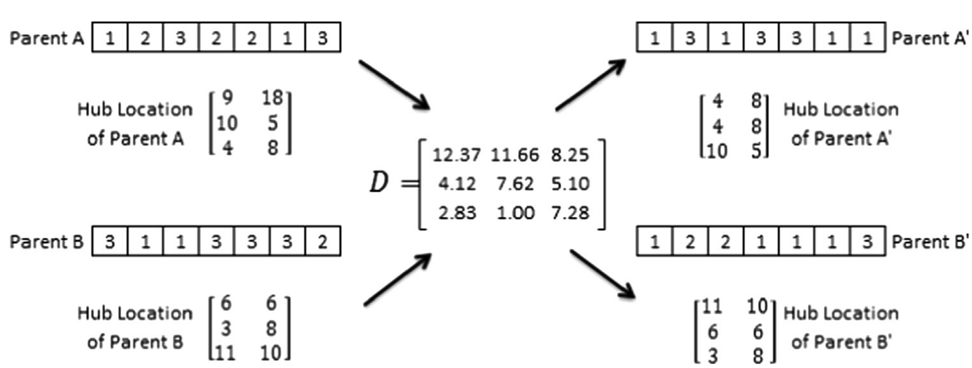

Fig. 2. An example of hub renumbering.

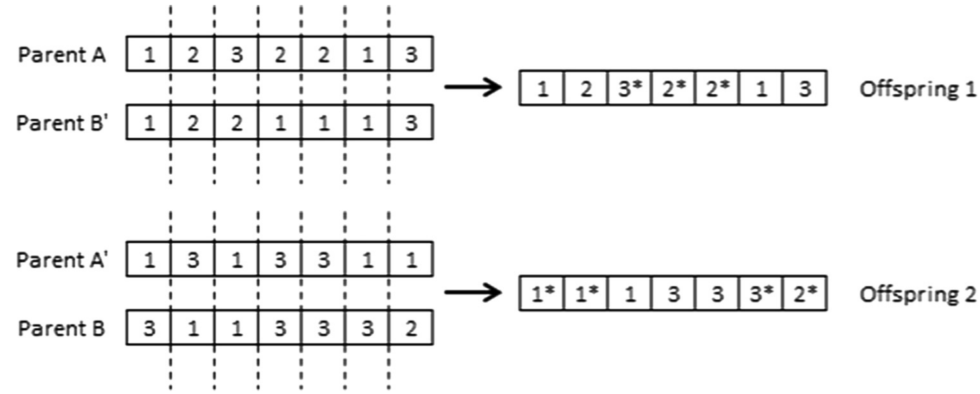

Fig. 3. An example of the assignment crossover (* represents the genes randomly selected from parents).

In the location operator, for given assignments, each hub location is updated by the Hyperbolic Approximation Procedure (HAP) proposed by Eyster et al. [18]. Aykin and Brown [14] also used this procedure to update facility locations when facilities interact with each other. Our mutation operator updates hub locations as in the following equation:

$$\mathbf{y}'_k = \frac{\sum_{l \neq k}\left(\dfrac{2\ transfer_{kl}\ \mathbf{y}_l}{d(\mathbf{y}_k, \mathbf{y}_l)^2}\right) + \sum_i \left(\dfrac{(collection_{ik} + distribution_{ki})\mathbf{y}_k}{d(\mathbf{y}_k, \mathbf{a}_i)^2}\right)}{\sum_{l \neq k}\left(\dfrac{2\ transfer_{kl}}{d(\mathbf{y}_k, \mathbf{y}_l)^2}\right) + \sum_i \left(\dfrac{(collection_{ik} + distribution_{ki})}{d(\mathbf{y}_k, \mathbf{a}_i)^2}\right)} \tag{16}$$

In Eq. (16), the operator denotes that the location of hub k depends on the locations of the spokes and other hubs. It is similar to the center update in the Weizsfeld Algorithm [19] where the new center is the convex combination of the points whose weights are calculated as the distances between the previous center and the points.

Both mutation operators can be used iteratively until convergence. However, in PHLGA, one step improvement is applied to each hub of the offspring. If an offspring is to be mutated, first the assignment operator is applied followed by the location operator. A mutated offspring replaces the original offspring only if it improves the fitness.

3.7. Generation replacement

Replacement is used to determine which individuals are going to survive in the next generation. In PHLGA, a kind of tournament selection is used based on the fitness function. After each reproduction, the first parent and two offspring join the tournament and the best fitted individual among them is favored. Since every parent is exposed to reproduction twice, once as the first parent and once as the second parent, we include only the first parent in the tournament to avoid premature convergence.

3.8. PHLGA parameters

Setting the parameters of a genetic algorithm is an important issue since these parameters may have a significant effect on the performance. These settings require extensive experimentation, therefore an algorithm with few parameters is favorable. PHLGA has only three parameters. Values of these parameters are determined as follows by analyzing preliminary run results.

- Population size is set as $\min\{np, 100\}$. The number of nodes and the number of hubs determine the complexity of a problem instance, and increasing the population size accordingly is a reasonable common practice. However, we limit the population size by 100 for large problems to keep the computation times at acceptable levels.
- Termination condition is determined as 300 generations. This is decided by plotting average and best fitness of the population over generations and examining the convergence behavior of the algorithm. We try to make sure that the algorithm converges to a good solution without wasting to much computation time.
- Mutation probability is selected as 0.2. In our preliminary runs we observed that lower values result in premature convergence and higher values increase the computation time without improving the solution quality.

The general framework of the algorithm is given in Algorithm 1, where p_A and p_B represent parents A and B, o_1 and o_2 stand for offspring 1 and 2.

Algorithm 1. PHLGA.

```
{p_1, …, p_pop-size} := GENERATE INITIAL POPULATION()
repeat
  r(.) := random permutation of 1, …, pop-size
  for i = 1 to pop-size do
    p_A := p_r(i) and p_B := p_r(i+1)
    (p'_A, p'_B) = RENUMBER HUBS (p_A, p_B)
    o_1 := CROSSOVER (p_A, p'_B)
    o_2 := CROSSOVER (p'_A, p_B)
    for j = 1 to 2 do
      p = rand(0, 1)
      if p ≤ mutation probability then
        o_j-mut := MUTATION (o_j)
        o_j := SELECT BEST (o_j-mut, o_j)
      end if
    end for
    SELECT BEST (o_1, o_2, p_A)
  end for
until Termination condition is satisfied
return The best individual
```

4. Computational results

In this section, we present the performance evaluation of PHLGA in terms of both solution quality and computation time. In Section 4.1, we compare PHLGA with Weiszfeld algorithm using simulated datasets that transform hub location problem to multi-facility location problem. We also compare results of PHLGA with DHLP solutions found by complete enumeration. We analyze the effects of problem parameters such as weights and transfer costs as well. In Section 4.2, we report our results on AP and CAB datasets and compare them with results of a location–allocation algorithm from the literature and with DHLP solutions.

PHLGA and complete enumeration were coded in MATLAB Version R2012a. We also recoded all existing algorithms used in comparisons in the same environment. Runs were made on a computer with Intel Core i7 CPU, 3.40 GHz processor and 8 GB RAM.

4.1. Results for simulated datasets

To the best of our knowledge, there is neither any dataset of which optimal solutions is known nor benchmark results of common datasets for PHLP. Therefore, we tested our algorithm on simulated datasets of which optimal hub locations can be found.

4.1.1. Comparison with optimal solution

Firstly, we generated datasets based on the relationship between MFLP and PHLP. If the discount factor (τ) and the unit cost parameter of transfer links (α_t) are taken as zero, and unit cost parameters of collection and distribution links (α_c and α_d) are the same, then PHLP transforms to MFLP. Under these cost settings, if we generate clustered nodes such that all clusters are well-separated from each other, the collection of Weiszfeld solutions [19] for individual clusters constitute the optimal solution of MFLP. Optimal facility locations in MFLP become optimal hub locations in PHLP. One example is shown in Fig. 4.

We used two different weight (w_{ij}) settings, unit weights and random weights uniformly distributed between 1 and 10. The characteristics of problem instances are given in Table 1.

We ran five replications of PHLGA for each problem instance given in Table 1. We report the Weiszfeld solution and the best PHLGA solution among five replications in Table 2 for all problem instances. We also give the respective DHLP solutions computed by a smartly designed complete enumeration algorithm for only small problem instances with 40 or fewer nodes.

According to Table 2, PHLGA solutions are almost the same as Weiszfeld solutions. As expected, working with PHLP results in lower total transportation cost than working with DHLP. When the number of nodes increases, the gap between the planar and the discrete solutions decreases. Since the density of spokes is higher in large problem instances the probability that locations of a hub and a spoke coincide increases. Hub locations become similar and the gap between planar and discrete solutions becomes smaller.

Computation times of the simulated problem instances are given in Table 3. According to Table 3, PHLGA solves problems in reasonable times. Even for the largest problem instance with 400 nodes, the solution time of a replicate is about 50 min. The assignments are given in the Weiszfeld solution, but PHLGA has to assign nodes to hubs in addition to locating hubs. Therefore, computation times of Weiszfeld algorithm and PHLGA are not comparable.

4.1.2. Effect of weight parameters

To analyze the effect of weight parameters (w_{ij}) representing the amount of flow between pairs of nodes, we generated a small problem instance with 10 nodes and two hubs. These nodes are divided into two equal-sized, well-separated groups as in Fig. 5(a). Unit cost parameters are set as $\alpha_c = \alpha_d = 1$ and $\alpha_t = 0.5$. When all weights are one initially, the discrete solution and the PHLGA solution are shown in Fig. 5(a). The two farthest nodes (i and j) are the spokes whose weights to each other will gradually be increased and eventually dominate the weights between other pairs of spokes.

We expect the hub locations to get closer to nodes i and j as $w_{ij} = w_{ji}$ increase. As seen in Fig. 5(b), the distance between hubs indeed increases continuously in the planar solution of PHLGA,

Table 1
Characteristics of simulated problem instances.

Problem	Number of		Weights
	Nodes	Hubs	
$P_{20,2,U}$	20	2	Unit
$P_{30,3,U}$	30	3	Unit
$P_{40,4,U}$	40	4	Unit
$P_{200,2,U}$	200	2	Unit
$P_{300,3,U}$	300	3	Unit
$P_{400,4,U}$	400	4	Unit
$P_{20,2,R}$	20	2	Uniform(1,10)
$P_{30,3,R}$	30	3	Uniform(1,10)
$P_{40,4,R}$	40	4	Uniform(1,10)
$P_{200,2,R}$	200	2	Uniform(1,10)
$P_{300,3,R}$	300	3	Uniform(1,10)
$P_{400,4,R}$	400	4	Uniform(1,10)

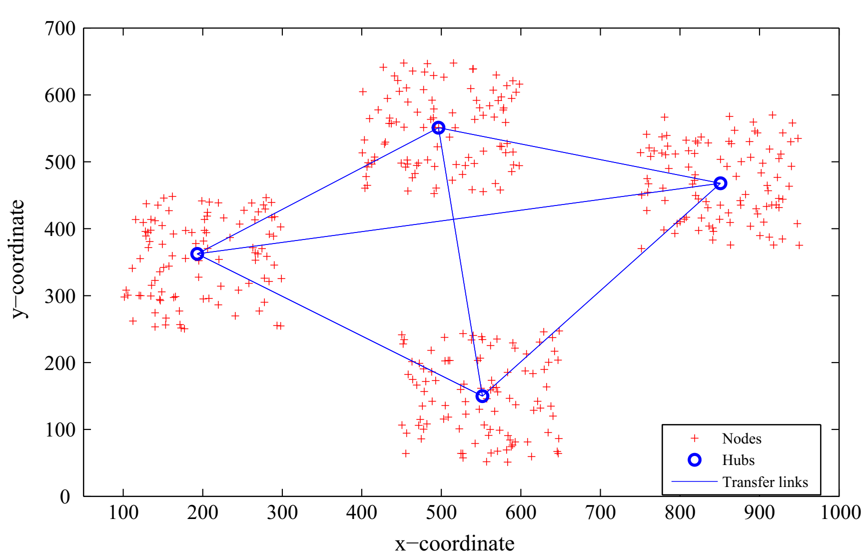

Fig. 4. Example simulated dataset for comparison with the Weiszfeld solution.

Table 2
Solution quality of PHLGA for simulated problem instances.

Problem instance	Objective function Value			% Deviations from Weiszfeld Solution	
	Weiszfeld	PHLGA[a]	Discrete	PHLGA	Discrete
$P_{20,2,U}$	5928.0927	5928.0928	6124.8124	0.0000	3.3184
$P_{30,3,U}$	11,613.7768	11,613.7849	11,875.8897	0.0001	2.2569
$P_{40,4,U}$	18,919.6466	18,919.6467	19,212.9879	0.0000	1.5505
$P_{200,2,U}$	6,023,955.0065	6,023,955.0066	–	0.0000	–
$P_{300,3,U}$	14,104,948.6421	14,104,948.6422	–	0.0000	–
$P_{400,4,U}$	24,807,083.6291	24,807,083.6295	–	0.0000	–
$P_{20,2,R}$	33,324.6079	33,324.6081	34,489.5794	0.0000	3.4958
$P_{30,3,R}$	64,257.6231	64,257.6233	65,741.5509	0.0000	2.3093
$P_{40,4,R}$	102,901.8776	102,901.8996	104,678.7668	0.0000	1.7268
$P_{200,2,R}$	33,120,562.6691	33,120,562.6691	–	0.0000	–
$P_{300,3,R}$	77,602,426.6896	77,602,426.6897	–	0.0000	–
$P_{400,4,R}$	136,372,901.7916	136,372,901.7924	–	0.0000	–

[a] Best of 5 runs.

Table 3
Computation times of PHLGA for simulated problem instances.

Problem instance	Computation time (s)	
	Weiszfeld	PHLGA[a]
$P_{20,2,U}$	0.0140	30.7447
$P_{20,2,U}$	0.0246	109.7641
$P_{20,2,U}$	0.0339	271.1754
$P_{20,2,U}$	0.4230	1076.9538
$P_{20,2,U}$	0.9365	1909.1330
$P_{20,2,U}$	1.6966	3030.3800
$P_{20,2,U}$	0.2953	29.6056
$P_{20,2,U}$	0.0292	109.4154
$P_{20,2,U}$	0.0462	271.5367
$P_{20,2,U}$	0.4260	1075.9186
$P_{20,2,U}$	0.9461	1907.5554
$P_{20,2,U}$	1.7523	3030.1109

[a] Average of 5 runs.

approaching the two farthest nodes. In the discrete solution, however, the distance is a stepwise function since a hub may only jump from one spoke to another. When weights of nodes i and j dominate the others, PHLGA locates the hubs on these nodes with small fractional errors.

4.1.3. Effect of cost parameters

In the case of well-separated node groups, as the discount factor and the difference between the unit cost parameters decreases, hubs are expected to get closer to each other. In order to see that, we generated a small problem instance with six nodes and two hubs as in Fig. 6(a). In this problem, weights are generated from uniform distribution between 1 and 10. We assumed that the unit collection and distribution cost parameters are $\alpha_c = \alpha_d = 1$. We started with a discount factor of zero and gradually increased it until it is one. Some selected solutions are shown in Fig. 6(a) as the transfer cost α_t changes. The change in the distance between hubs is plotted for PHLP and discrete solutions in Fig. 6(b).

As expected, PHLGA finds closer hubs as the discount factor decreases, and the transfer cost gets closer to the collection and distribution costs. As seen in Fig. 6(b), PHLP is more sensitive than DHLP to the changes in cost parameters. While the distance between hubs decreases as a step-wise function in DHLP, hubs get closer steadily in PHLP.

4.2. Results for real world datasets

There are two common datasets in the literature for DHLP, namely Australian Post (AP) and Civil Aerospace Board (CAB) datasets. In DHLP instances, coordinates of spokes need not be known because the distance parameters (d_{ij}) are sufficient to calculate the objective function value. For PHLP we need the coordinates, therefore there are no commonly used datasets for PHLP. However, Ernst and Krishnamoorthy [6] share the AP dataset that includes coordinates of nodes. From this dataset, we generated a number of problem instances with different sizes. The second dataset CAB is first introduced by O'Kelly in 1987 [2]. This dataset includes 25 cities. We had to extract location of nodes by using a jpeg file of the CAB dataset, because we could not find exact coordinates of cities.

We compared PHLGA solutions with discrete solutions obtained by our enumeration algorithm and the location–allocation algorithm proposed to solve PHLP by [14] which we recoded.

We first report the results on the AP dataset. Problems generated from this dataset were solved for 2-hub, 3-hub and 4-hub cases with given location, weight, and cost parameters ($\alpha_c = 3$, $\alpha_d = 2$, and $\alpha_t = 0.75$). Because of the time constraint, discrete solutions for problems with 100 and 200 nodes were not calculated. The solution quality and computation time results are given in Tables 4 and 5, respectively. $P_{n,p}$ in these tables indicates a problem instance with n nodes and p hubs.

According to Table 4, PHLGA solutions are up to 1.3 percent better than the discrete solutions, where the average is only 0.4 percent for small problems. Moreover, PHLGA dominates the location-allocation algorithm. Although the location-allocation algorithm is faster than PHLGA according to Table 5, the average solution quality of this algorithm is 1.7 percent worse than PHLGA. Aykin and Brown's algorithm is based on a common location-allocation procedure, and it may get stuck at a local optimum. In PHLGA, both the crossover and the mutation operators facilitate better exploration of the solution space. This prevents the algorithm from getting stuck at local optima. Although computation times of PHLGA are longer than the location-allocation algorithm, the longest time for the 200-node and 4-hub problem is approximately 15 min, which is quite reasonable for a genetic algorithm.

We also solved the same AP problem instances using unit weights to see the direct effect of distances. We do not report the computation times since they are similar to those in Table 5. The solution quality results are given in Table 6. Again, PHLGA solutions are better than both discrete and location-allocation algorithm solutions. When the weights of some spokes are larger as in

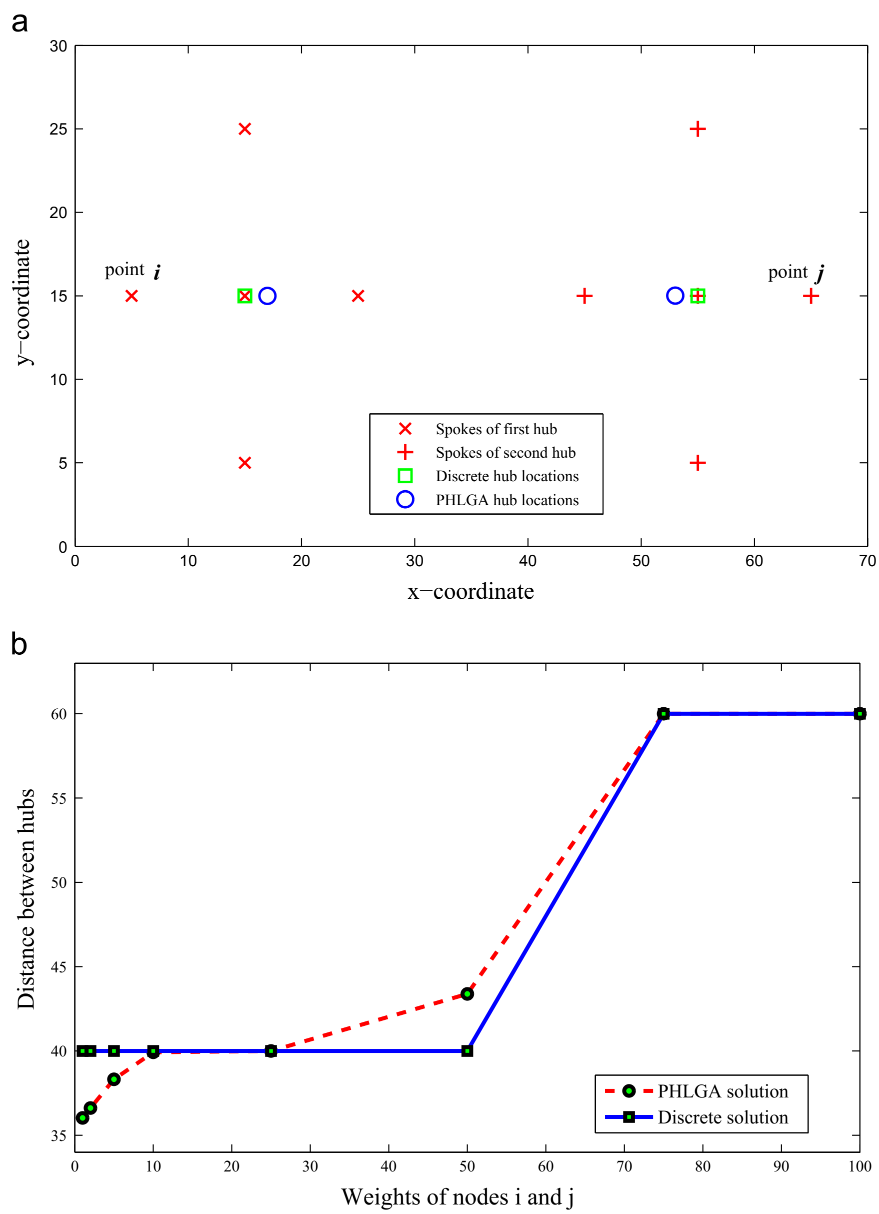

Fig. 5. Example problem to analyze effect of weight parameters: (a) PHLGA and discrete solutions with unit weights and (b) distance between hubs as $w_{ij}=w_{ji}$ increase.

the original case, PHLGA is forced to locate hubs nearby these spokes. With unit weights, however, the distance alone dictates the hub locations. Therefore, the average gap between discrete and PHLGA solutions in the unit weight case is slightly larger (0.6 percent) than that in the original weight case. The gap between the location-allocation algorithm and PHLGA solutions in the unit weight case is about the same (1.6 percent) as in the original weight case.

For the CAB dataset, we used given weights and coordinates extracted from the jpeg file. This dataset contains only 25 cities, therefore we could experiment by changing the transfer cost parameter. While the collection and distribution cost parameters are fixed as $\alpha_c=\alpha_d=1$, the unit transfer cost α_t is increased from 0 to 1 with increments of 0.25. We solved problem for 2-hub, 3-hub and 4-hub. Solution quality and computation times are given in Tables 7 and 8, respectively. P_{n,p,α_t} in the tables indicates a problem instance with n nodes, p hubs, and transfer cost α_t.

For all transfer cost values in Table 7, PHLGA solutions are better than discrete solutions, as expected. PHLGA also surpasses location-allocation algorithm for all transfer cost values. Therefore, our algorithm is robust to changes in the transfer cost parameter. According to Table 8, PHLGA solves a 25-node and 4-hub problem in approximately 1 minute. We observe that, as the number of hubs increases, computation times also increase as expected.

CAB problem instances were also solved with unit weights. The solution quality results are given in Table 9. PHLGA solutions are again better than both discrete and location-allocation solutions, with the exception of two problems for which discrete solutions are slightly better.

4.3. Insight for DHLP

PHLP solutions may provide valuable insight for DHLP. For this purpose, a modified version of PHLGA, PHLGA-D, can be used to find benchmark solutions for large problem instances of DHLP. In every generation of the algorithm, PHLGA-D takes the hubs in the best PHLGA solution and simply carries them to the locations

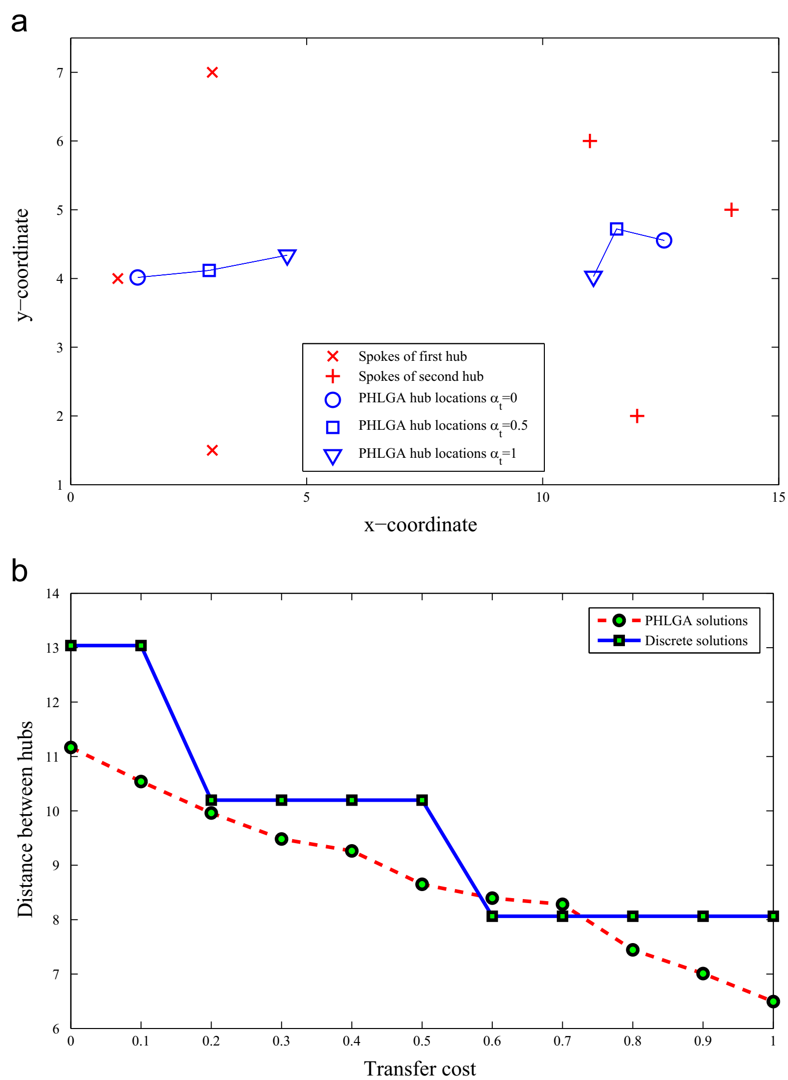

Fig. 6. Example problem to analyze effect of cost parameters: (a) PHLGA solutions with different transfer costs and (b) distance between hubs as α_t increases.

Table 4
Comparison of solution quality for AP problem instances with given weights

Problem instance	Objective function Value			% Deviation from PHLGA	
	PHLGA[a]	Discrete	Location–allocation[b]	Discrete	Location–allocation
$P_{10,2}$	144,454,712.5984	146,431,653.1919	145,948,318.5361	1.3686	1.0340
$P_{10,3}$	118,822,719.1562	118,822,719.1562	118,822,723.7561	0.0000	0.0000
$P_{10,4}$	99,542,115.6300	99,542,115.6255	100,807,994.3014	0.0000	1.2717
$P_{20,2}$	161,108,872.7467	161,123,549.8212	162,919,134.9659	0.0091	1.1236
$P_{20,3}$	140,190,014.6141	141,518,313.6022	145,407,154.4272	0.9475	3.7215
$P_{20,4}$	127,468,585.2267	128,783,868.5636	128,072,700.7163	1.0318	0.4739
$P_{25,2}$	165,338,355.0426	165,526,106.7494	167,015,324.5975	0.1136	1.0143
$P_{25,3}$	146,553,657.0823	146,664,703.1743	147,834,697.7674	0.0758	0.8741
$P_{25,4}$	131,650,147.4725	132,466,065.7684	132,321,276.0737	0.6198	0.5098
$P_{50,2}$	172,974,101.5434	173,110,087.5033	175,021,291.7846	0.0786	1.1835
$P_{50,3}$	153,842,188.6623	153,875,860.8138	156,620,256.8029	0.0219	1.8058
$P_{50,4}$	139,356,424.5933	139,384,991.4557	148,310,221.9366	0.0205	6.4251
$P_{100,2}$	176,859,703.5806	-	177,864,735.2402	–	0.5683
$P_{100,3}$	157,822,997.1343	-	160,487,288.7296	–	1.6882
$P_{100,4}$	143,101,592.9489	-	151,466,894.0157	–	5.8457
$P_{200,2}$	182,399,818.0866	-	183,022,158.9641	–	0.3412
$P_{200,3}$	162,743,687.1816	-	164,754,626.6326	–	1.2356
$P_{200,4}$	147,357,809.9290	-	149,618,674.7463	–	1.5343

[a] Best of 5 runs.
[b] Best of 10 runs.

of the closest spokes. After that, spokes are reassigned to new (discrete) hubs by using equation (12).

The computation time of this additional operation is negligible, therefore we do not report computation times of PHLGA-D. In Table 10, we present PHLGA-D solutions for the AP dataset. According to the table, PHLGA-D finds the same solutions as complete enumeration most of the time. There are only two problem instances for which PHLGA-D and enumeration solutions are barely different.

Table 11 summarizes PHLGA-D results for the CAB dataset. As for the AP dataset, PHLGA-D finds the same solutions as complete enumeration for most problems. The percent deviations are acceptable for the problem instances for which PHLGA-D and complete enumeration results are different.

In summary, PHLGA-D finds very good solutions for DHLP in reasonable times. Its solutions for large problem instances can be used as a benchmark.

5. Conclusion

We study the uncapacitated single allocation planar hub location problem. We provide a mathematical formulation of the PHLP, discuss the main differences of PHLP from the discrete version of the problem, and propose a genetic algorithm for solving PHLP. Distinguishing features of PHLGA are: (1) it is based on two new crossover operators for node assignment and hub location, (2) it uses two new mutation operators for these two decisions, and (3) it has only three parameters to be set.

Computational results of the algorithm are presented on both simulated and real world datasets. The solution quality of PHLGA is quite satisfactory. It yields expected results for simulated datasets as problem parameters are changed systematically. It also finds better solutions for real world datasets than location-allocation algorithm and discrete solutions. To the best of our knowledge, there are no benchmark studies for PHLP and we present the first results for well-known real world datasets. Although the computation times of PHLGA are longer than the other algorithms, they are within reasonable limits. Also, since hub location decisions are strategic and long term, solution times of large instances are acceptable.

PHLP is underappreciated in literature compared to the discrete version of the problem. Because of the continuous nature of the problem, PHLP is more sensitive to problem parameters than DHLP. Analyzing the effects of these parameters and using the PHLP solutions, one can provide valuable insight for the discrete version problem.

In the future, we plan to work on different variants of PHLP such as the multiple allocation version and the capacitated case. One can concentrate on different search strategies, particularly for locating hubs in the continuous space. Other heuristic and meta-heuristic approaches can also be developed for solving PHLP.

Table 5
Comparison of computation times for AP problem instances with given weights.

Problem instance	Computation time (s)	
	PHLGA[a]	Location–allocation[b]
$P_{10,2}$	8.4882	0.0706
$P_{10,3}$	14.2972	0.0643
$P_{10,4}$	21.8101	0.1482
$P_{20,2}$	30.1118	0.5098
$P_{20,3}$	51.3619	0.8634
$P_{20,4}$	78.1652	0.3617
$P_{25,2}$	46.6801	0.0518
$P_{25,3}$	80.9423	0.1530
$P_{25,4}$	123.1486	2.3591
$P_{50,2}$	175.5619	0.3487
$P_{50,3}$	198.8619	0.3337
$P_{50,4}$	225.3200	0.3403
$P_{100,2}$	343.1047	0.7121
$P_{100,3}$	391.2937	1.6203
$P_{100,4}$	441.7569	1.2826
$P_{200,2}$	717.6016	7.8417
$P_{200,3}$	819.0681	1.5234
$P_{200,4}$	917.4598	2.2601

[a] Average of 5 runs.
[b] Average of 10 runs.

Table 6
Comparison of solution quality for AP problem instances with unit weights.

Problem instance	Objective function value			% Deviation from PHLGA	
	PHLGA[a]	Discrete	Location–allocation[b]	Discrete	Location–allocation
$P_{10,2}$	4,706,233.9454	4,748,513.6437	4,750,014.3682	0.8984	0.9303
$P_{10,3}$	3,854,382.9401	3,858,434.3394	3,867,779.1799	0.1051	0.3476
$P_{10,4}$	3,275,897.2525	3,277,537.4675	3,305,438.1810	0.0501	0.9018
$P_{20,2}$	2,186,0426.4948	21,898,523.3065	21,898,523.5439	0.1743	0.1743
$P_{20,3}$	18,908,588.6573	19,050,448.4556	19,241,326.7778	0.7502	1.7597
$P_{20,4}$	17,094,834.2209	17,226,106.7330	17,644,457.9940	0.7679	3.2151
$P_{25,2}$	36,053,131.1298	36,331,769.8130	36,393,212.5263	0.7729	0.9433
$P_{25,3}$	31,393,226.4248	31,658,947.5902	31,658,052.9799	0.8464	0.8436
$P_{25,4}$	27,894,936.2313	28,240,863.6645	28,093,450.5573	1.2401	0.7117
$P_{50,2}$	151,405,544.3708	152,434,435.0635	155,604,645.7397	0.6796	2.7734
$P_{50,3}$	133,154,419.1975	134,430,796.4648	135,561,663.5860	0.9586	1.8079
$P_{50,4}$	119,783,376.6138	120,617,622.1492	122,918,783.3630	0.0339	2.6176
$P_{100,2}$	630,556,752.1394	–	636,349,329.1788	–	0.9186
$P_{100,3}$	557,774,747.5249	–	568,946,816.2098	–	2.0030
$P_{100,4}$	507,165,725.9625	-	516,794,545.9969	–	1.8986
$P_{200,2}$	2,562,301,859.9803	-	2,596,100,858.9048	–	1.3191
$P_{200,3}$	2,272,722,667.3472	-	2,309,558,338.1396	–	1.6208
$P_{200,4}$	2,080,237,102.0118	-	2,150,641,231.1835	–	3.3844

[a] Best of 5 runs.
[b] Best of 10 runs.

Table 7
Comparison of solution quality for CAB problem instances with given weights.

Problem instance	Objective function Value			% Deviation from PHLGA	
	PHLGA[a]	Discrete	Location–allocation[b]	Discrete	Location–allocation
$P_{25,2,0}$	1,846,858,691.6409	1,851,519,882.6617	1,846,860,050.8699	0.2524	0.0001
$P_{25,3,0}$	1,238,841,968.6077	1,250,799,876.4639	1,239,442,280.7378	0.9652	0.0485
$P_{25,4,0}$	926,139,770.7348	926,139,727.5804	926,142,361.4224	0.0000	0.0003
$P_{25,2,0.25}$	2,117,245,526.0451	2,125,213,449.0519	2,119,580,873.3056	0.3763	0.1103
$P_{25,3,0.25}$	1,616,568,011.6486	1,623,355,808.2215	1,621,434,451.9774	0.4199	0.3010
$P_{25,4,0.25}$	1,365,638,218.6226	1,368,344,668.9857	1,368,346,875.3873	0.1982	0.1983
$P_{25,2,0.5}$	2,377,572,145.5770	2,398,907,015.4421	2,392,301,695.7408	0.8973	0.6195
$P_{25,3,0.5}$	1,973,105,443.0887	1,981,515,911.8951	1,989,774,105.6959	0.4263	0.8448
$P_{25,4,0.5}$	1,777,485,008.8324	1,783,299,985.9242	1,790,312,206.9453	0.3271	0.7216
$P_{25,2,0.75}$	2,620,316,820.2115	2,650,079,597.5970	2,646,605,953.7843	1.1358	1.0033
$P_{25,3,0.75}$	2,307,915,902.0199	2,332,468,346.8800	2,358,113,759.3768	1.0638	2.1750
$P_{25,4,0.75}$	2,170,751,744.0820	2,182,753,575.1116	2,206,388,156.7255	0.5529	1.6417
$P_{25,2,1}$	2,828,602,989.7345	2,851,417,617.5671	2,895,572,865.4553	0.8066	2.3676
$P_{25,3,1}$	2,608,781,735.2524	2,619,441,176.6997	2,742,266,725.0550	0.4086	5.1168
$P_{25,4,1}$	2,510,757,585.6315	2,528,699,239.8440	2,622,464,106.5095	0.7146	4.4491

[a] Best of 5 runs.
[b] Best of 100 runs.

Table 8
Comparison of computation times for CAB problem instances with given weights.

Problem instance	Computation time (seconds)	
	PHLGA[a]	Location–allocation[b]
$P_{25,2,0}$	44.6682	0.1210
$P_{25,3,0}$	50.7963	0.1709
$P_{25,4,0}$	59.5412	0.2157
$P_{25,2,0.25}$	44.6798	0.1165
$P_{25,3,0.25}$	50.8704	0.1129
$P_{25,4,0.25}$	62.2696	0.1722
$P_{25,2,0.5}$	45.5472	0.0985
$P_{25,3,0.5}$	52.0370	0.1531
$P_{25,4,0.5}$	61.1689	0.1838
$P_{25,2,0.75}$	45.6403	0.0622
$P_{25,3,0.75}$	51.8743	0.2687
$P_{25,4,0.75}$	61.8410	0.3181
$P_{25,2,1}$	45.6694	0.0618
$P_{25,3,1}$	52.0247	0.0929
$P_{25,4,1}$	60.2938	0.2104

[a] Average of 5 runs.
[b] Average of 100 runs.

Table 9
Comparison of solution quality for CAB problem instances with unit weights.

Problem instance	Objective function value			% Deviation from PHLGA	
	PHLGA[a]	Discrete	Location–allocation[b]	Discrete	Location–allocation
$P_{25,2,0}$	143,282.3656	144,715.5937	143,282.3656	1.0003	0.0000
$P_{25,3,0}$	106,178.1293	108,361.8780	106,178.1408	2.0567	0.0000
$P_{25,4,0}$	880,92.6106	89,989.3793	88,092.6253	2.1532	0.0000
$P_{25,2,0.25}$	164,914.5774	167,868.0798	166,098.8594	1.7909	0.7181
$P_{25,3,0.25}$	135,868.9821	137,091.6587	138,383.8149	0.8999	1.8509
$P_{25,4,0.25}$	121,307.2666	123,331.5755	123,141.3501	1.6687	1.5119
$P_{25,2,0.5}$	183,690.7567	184,919.1342	188,259.2612	0.6687	2.4871
$P_{25,3,0.5}$	162,287.6038	163,584.4113	168,260.5242	0.7991	3.6805
$P_{25,4,0.5}$	150,659.2110	152,316.7232	159,334.5833	1.1002	5.7583
$P_{25,2,0.75}$	197,843.1092	199,143.3822	208,416.3639	0.6572	5.3443
$P_{25,3,0.75}$	183,521.4050	184,823.7168	199,268.2199	0.7096	8.5804
$P_{25,4,0.75}$	176,187.3259	174,917.7062	190,876.3816	−0.7206	8.3372
$P_{25,2,1}$	207,258.0130	207,257.8957	217,547.6863	−0.0001	4.9647
$P_{25,3,1}$	201,412.9561	201,604.7906	216,782.3151	0.0952	7.6308
$P_{25,4,1}$	195,755.9558	196,181.7640	213,600.7094	0.2175	9.1158

[a] Best of 5 runs.
[b] Best of 100 runs.

Table 10
Solution quality of PHLGA-D for AP problem instances with given weights.

Problem instance	Objective function value		% Deviation of PHLGA-D from discrete
	PHLGA-D	Discrete	
$P_{10,2}$	146,431,653.1919	146,431,653.1919	0.0000
$P_{10,3}$	118,822,719.1562	118,822,719.1562	0.0000
$P_{10,4}$	99,542,115.6255	99,542,115.6255	0.0000
$P_{20,2}$	161,123,549.8212	161,123,549.8212	0.0000
$P_{20,3}$	141,518,313.6022	141,518,313.6022	0.0000
$P_{20,4}$	128,783,868.5636	128,783,868.5636	0.0000
$P_{25,2}$	165,658,455.8055	165,526,106.7494	0.0800
$P_{25,3}$	146,664,703.1743	146,664,703.1743	0.0000
$P_{25,4}$	132,616,529.9470	132,466,065.7684	0.1136
$P_{50,2}$	173,110,087.5033	173,110,087.5033	0.0000
$P_{50,3}$	153,875,860.8138	153,875,860.8138	0.0000
$P_{50,4}$	139,384,991.4557	139,384,991.4557	0.0000
$P_{100,2}$	176,906,694.8383	–	–
$P_{100,3}$	157,947,223.4856	–	–
$P_{100,4}$	143,491,289.5440	–	–
$P_{200,2}$	180,395,483.2135	–	–
$P_{200,3}$	161,044,099.0967	–	–
$P_{200,4}$	146,617,823.8019	–	–

Table 11
Solution quality of PHLGA-D for CAB problem instances with given weights.

Problem instance	Objective function value		% Deviation of PHLGA-D from discrete
	PHLGA-D	Discrete	
$P_{25,2,0}$	1,851,519,882.6617	1,851,519,882.6617	0.0000
$P_{25,3,0}$	1,257,767,894.3865	1,250,799,876.4639	0.5571
$P_{25,4,0}$	926,139,727.5804	926,139,727.5804	0.0000
$P_{25,2,0.25}$	2,125,213,449.0519	2,125,213,449.0519	0.0000
$P_{25,3,0.25}$	1,623,355,808.2215	1,623,355,808.2215	0.0000
$P_{25,4,0.25}$	1,368,344,668.9857	1,368,344,668.9857	0.0000
$P_{25,2,0.5}$	2,398,907,015.4421	2,398,907,015.4421	0.0000
$P_{25,3,0.5}$	1,981,515,911.8951	1,981,515,911.8951	0.0000
$P_{25,4,0.5}$	1,783,299,985.9242	1,783,299,985.9242	0.0000
$P_{25,2,0.75}$	2,661,475,015.6487	2,650,079,597.5970	0.4300
$P_{25,3,0.75}$	2,332,468,346.8800	2,332,468,346.8800	0.0000
$P_{25,4,0.75}$	2,182,753,575.1116	2,182,753,575.1116	0.0000
$P_{25,2,1}$	2,851,417,617.5671	2,851,417,617.5671	0.0000
$P_{25,3,1}$	2,637,191,896.9972	2,619,441,176.6997	0.6777
$P_{25,4,1}$	2,555,255,741.4709	2,528,699,239.8440	1.0502

In addition, the relationship between the continuous and the discrete versions of the problem can be used to adapt PHLP solution procedures for DHLP and vice versa. Another future research area is to borrow more ideas and solution procedures from the multi-facility location literature.

References

[1] Alev SA, Kara BY. Network hub location problems: the state of art. Eur J Oper Res 2008;902:1–21. http://dx.doi.org/10.1016/j.ejor.2007.06.008.

[2] Me O. A quadratic integer program for the location of interacting hub facilities. Eur J Oper Res 1987;32:393–404. doi: http://dx.doi.org/10.1016/S0377-2217(87)80007-3.

[3] Klincewisz JG. Heuristics for the p-hub location problem. Eur J Oper Res 1991;53:25–37. http://dx.doi.org/10.1016/0377-2217(91)90090-I.

[4] Klincewisz JG. Avoiding local optima in the p-hub location problem using tabu search and grasp. Ann Oper Res 1992;40:283–302. http://dx.doi.org/10.1007/BF02060483.

[5] Campbell JF. Hub location and the p-hub median problem. Oper Res 1996;44:923–35 URL: ⟨http://www.jstor.org/stable/171583.

[6] Ernst AT, Krishnamoorthy M. Efficient algorithms for the uncapacitated single allocation p-hub median problem. Locat Sci 1996;4:139–54. http://dx.doi.org/10.1016/S0966-8349(96)00011-3.

[7] Abdinnour-Helm S. Using simulated annealing to solve the p-hub median problem. Int J Phys Distrib Logist Manage 2001;31:203–20. http://dx.doi.org/10.1108/09600030110389532.

[8] Ernst AT, Krishnamoorthy M. Exact and heuristic algorithms for the uncapacitated multiple allocation p-hub median problem. Eur J Oper Res 1998;104:100–12. http://dx.doi.org/10.1016/S0377-2217(96)00340-2.

[9] Topcuoglu H, Corut F, Ermis M, Yilmaz G. Solving the uncapacitated hub location problem using genetic algorithms. Comput Oper Res 2005;32:967–84. http://dx.doi.org/10.1016/j.cor.2003.09.008.

[10] Kratica J, Stanimirovic Z, Tosic D, Filipovic V. Two genetic algorithms for solving the uncapacitated single allocation p-hub median problem. Eur J Oper Res 2007;182:15–28. http://dx.doi.org/10.1016/j.ejor.2006.06.056.

[11] Ilic A, Urosevic D, Brimberg J, Mladenovic N. A general variable neighborhood search for solving the uncapacitated single allocation p-hub median problem. Eur J Oper Res 2010;36:289–300. http://dx.doi.org/10.1016/j.ejor.2010.02.022.

[12] O'Kelly ME. A clustering approach to the planar hub location problem. Ann Oper Res 1992;40:339–53. http://dx.doi.org/10.1007/BF02060486.

[13] Campbell JF. Locating transportation terminals to serve an expanding demand. Transp Res Part B: Methodol 1990;24:173–92. http://dx.doi.org/10.1016/0191-2615(90)90015-Q.

[14] Aykin T, Brown GF. Interacting new facilities and location–allocation problems. Transp Sci 1992;26:212–22. http://dx.doi.org/10.1287/trsc.26.3.212.

[15] Campbell JF, O'Kelly ME. Twenty-five years of hub location research. Transp Sci 2012;46:153–69. http://dx.doi.org/10.1287/trsc.1120.0410.

[16] Megiddo N, Supowit KJ. On the complexity of some common geometric location problems. SIAM J Comput 1984;13:182–96. http://dx.doi.org/10.1137/0213014.

[17] Falkenauer E. Genetic algorithms and grouping problems. New York: Wiley; 1998.

[18] Eyster JW, White JA, Wierwille WW. On solving multifacility location problems using a hyperboloid approximation procedure. Am Inst Ind Eng Trans 1973;5:1–6 URL: 〈http://www.jstor.org/stable/171609〉.

[19] Weiszfeld E. Sur le point par lequel la somme des distances de n points donnés est minimum. Tohoku Math J 1937;43:355–86. http://dx.doi.org/10.1007/s10479-008-0352-z.