applied sciences

MDPI

Article

Multi-Scenario Cooperative Evolutionary Algorithm for the β-Robust p-Median Problem with Demand Uncertainty

Jianmai Shi [1,*], Xiaolei Zheng [2], Bo Jiao [1] and Rui Wang [1]

[1] Science and Technology on Information Systems Engineering Laboratory, College of System Engineering, National University of Defense Technology, Changsha 410073, China; jiaobonudt116@163.com (B.J.); ruiwangnudt@gmail.com (R.W.)

[2] National Defense University, Beijing 100091, China; frostday1023@163.com

* Correspondence: jianmaishi@gmail.com

Received: 18 September 2019; Accepted: 30 September 2019; Published: 5 October 2019

Abstract: In this paper, we studied the solution approach for the β-robust p-median problem with a large number of scenarios for the uncertain demands. The concept of neighborhood scenarios was introduced to describe the scenarios with a higher similarity than others. By utilizing knowledge from the solutions of neighborhood scenarios and the parallel search strategy, a novel multi-scenario cooperative evolutionary algorithm was proposed to solve the problem for all scenarios in one run. The proposed algorithm was compared with the widely used location–allocation heuristic and genetic algorithm through two practical cases, which were a network with 95 cities and a network with 668 demand nodes in an urban area. The computational results indicate that our algorithm can obtain better solutions in a much shorter time.

Keywords: facility location; metaheuristic; robust; uncertain demand; scenario

1. Introduction

The p-median problem is one of the most well-studied facility location problems in the field of management and operation research. In the classical p-median problem, a certain number of p facilities are located among n candidate locations and m demand points are allocated to the p opened facilities, while the objective is to minimize the sum of the transportation cost/distance to serve all of the demand points. The p-median problem and its extensions have been widely used in many practical situations including the location of plants, warehouses, distribution centers, hubs, and public service facilities [1]. Previous study has proven that the p-median problem is Non-deterministic Polynomial (NP) hard [2], and that its optimal solution is unlikely to be obtained through an exact approach with polynomial complexity. Thus, various approximate heuristics have been developed to find optimal or near optimal solutions.

The genetic algorithm (GA) is a kind of metaheuristic inspired by the biological evolution process, which is one of the most efficient metaheuristics for solving the p-median problem. Hosage and Goodchild [3] first reported the application of the GA for the p-median problem, and showed the advantage of the GA in the generality for approaching difficult optimization problems. Alp et al. [4] improved the efficiency of the GA for solving the p-median problem by integrating the greedy heuristic in the evolution process, which showed better performance than the simulated annealing algorithm proposed by Chiyoshi and Galvão [5]. Correa et al. [6] employed a heuristic "hypermutation" operator in the GA to solve the capacitated p-median problem, which obtained better results from a comparison with the Tabu search algorithm. Alba and Dominguez [7] compared a class of GA based heuristics

with several other metaheuristics and reported their relative performances. Their experimental results indicated that the GA based algorithm had advantages of wide applicability, low implementation effort, and high accuracy. Kratica et al. [8] proposed two GAs with new coding schemes and genetic operators for solving the single allocation p-hub median problem. Li et al. [9] investigated the impacts of initial population generating strategies on the performance of the GA, and proposed an approach to enhance the quality of the initial population. Experimental results showed that the computational time could be reduced by improving the initial population. Herda [10] introduced the parallel strategy in the GA to improve the efficiency in solving the capacitated p-median problem, and designed two parallel GAs that could run on computing clusters with high performance. Memari et al. [11] proposed a modified firefly algorithm, which was used to solve a supply chain network problem. GA has also been used to efficiently solve many other problems in the industrial field such as material handling control [12], the transportation–production problem [13], and the works transport organization and control problem [14]. More metaheuristics in the facility location field can be found in Mladenović et al. [1].

The p-median facility location problem is usually a long-term strategic decision problem, and most of the above studies assume that the facilities' operational environments is deterministic. However, in practical situations, the fierce competition, changes in customer taste, community growth, and economic vitality in the market area all make the demand quite uncertain in the planning period and difficult to estimate. In this situation, the demand is assumed to be an uncertain parameter, which can be modeled as a set of alternative future scenarios. Then, robust optimization approaches can be utilized to minimize the maximum regret of the overall transportation cost [15]. Weaver and Church [16] studied the p-median problem in a stochastic network, where the demand in each node was modeled as a set of discrete scenarios. Chen et al. [17] investigated the uncertain p-median problem, and developed an α-reliable mean-excess model to minimize the expected regret. Snyder and Daskin [18] introduced the β-robust approach for the p-median problem with uncertainties, which can be used to obtain a trade-off between the expected transportation cost/distance and the robustness. The β-robust approach for facility location integrates the advantages of both the stochastic and robust optimization methods. Then, Lim and Sonmez [19] further studied the β-robust facility relocation problem under budget constraint. Sakalli and Atabas [20] employed both ant colony optimization and the GA to solve a tactical production–distribution planning problem in an uncertain environment. In most of the studies on robust facility location, in order to estimate the regrets under different scenarios, the optimal solutions for all scenarios should be first obtained. The instance under each scenario usually corresponds to a kind of classical facility location problem, which is NP-hard. The solving of each scenario problem has to employ approximate heuristics such as the GA and Lagrangian heuristic [21], which is time-consuming. When the number of scenarios is lower, all of the instances can be solved through iteratively running an approximate heuristic. For example, nine scenarios were included in the experiments of both [18,19]. However, in many practical situations, the uncertainties have to be modeled as a larger number of scenarios, which means that the computational time would be unaffordable by iteratively running traditional methods.

Motivated by the requirement of efficiently solving the β-robust p-median problem with a large number of possible scenarios, a novel multi-scenario cooperative evolutionary (MSCE) metaheuristic was proposed. The definition of the neighborhood scenario was introduced, which was used to describe scenarios similar to each other. The major contribution of the MSCE algorithm is to utilize the knowledge obtained in the solutions of neighborhood scenarios. As all of the scenarios are used to model the uncertainties in a p-median problem, there are some similarities between these scenarios. The basic observation is that the optimal solution of a scenario is close to another scenario, if these two scenarios have higher similarities. Thus, the knowledge obtained in the solving of a scenario is helpful to the solving of its neighborhood scenarios. There are two advantages of the MSCE algorithm compared to other approximate heuristics. First, all scenarios are solved parallelly in one run of the MSCE algorithm, which improves the efficiency of computational time. Second, in the parallel search

process of MSCE, helpful knowledge is exchanged among neighborhood scenarios for a certain number of iterations, which improves the performance in both optimality and efficiency.

The rest of this paper is organized as follows. The model methodology is presented in Section 2. Section 3 proposes the MSCE algorithm and Section 4 presents the computational experiments. Finally, the paper is concluded in Section 5.

2. Model Methodology

In this section, a 0–1 integer programming formulation for the β-robust p-median problem is presented. The objective of the problem is to locate the number of p facilities required to minimize the overall expected transportation cost for satisfying all of the demand, subject to the constraints that the relative regret under each scenario is no more than β. In order to calculate the relative regret of a scenario, the optimal solution under this scenario should first be obtained through solving a corresponding deterministic p-median problem. The detail formulation is presented in the following subsections.

2.1. Notations

The main parameters and variables used in the model development are presented as follows.

Indices:	
M	Set of customer nodes, indexed by i;
N	Set of potential facilities, indexed by j;
S	Set of scenarios, indexed by s;
Parameters:	
c_{ij}	the cost of supplying one unit to customer node i from facility j;
σ_{is}	the demand of customer point i under scenario s;
p	the number of facilities that can be opened;
q_s	the probability that scenario s may occur;
β	a constant that represents the maximum relative regret permitted for each scenario;
z_s^*	the optimal objective function value of the problem under scenario s;
Decision variables:	
Y_j	is 1, if facility j is opened in the robust optimization problem; otherwise 0;
Y_{js}	is 1, if facility j is opened in the problem under scenario s; otherwise 0;
X_{ijs}	is 1, if the demand of customer node i is supplied by facility j under scenario s; otherwise 0.

2.2. Subproblem Formulation for a Given Scenario

Under a given scenario, the problem is a classical p-median facility location problem, which can be formulated as follows.

$$P_s: \ Minz_s = \sum_{i \in M} \sum_{j \in N} \sigma_{is} c_{ij} X_{ijs} \tag{1}$$

Subject to

$$\sum_{j \in N} Y_{js} = p \tag{2}$$

$$X_{ijs} \le Y_{js}, \ \forall i \in M, \ j \in N, \tag{3}$$

$$\sum_{j \in N} X_{ijs} = 1, \ \forall i \in M, \tag{4}$$

$$X_{ijs}, \ Y_{js} \in \{0, 1\}, \ \forall i \in M, \ j \in N. \tag{5}$$

Objective function (1) is to minimize the total transportation cost. In Equation (2), the number of opened facilities under scenario s is restricted to p. Constraint (3) ensures that the demand of the customer node i can only be supplied by opened facilities. Constraint (4) ensures that each customer node must be allocated to a facility. Constraint (5) is a standard integrality constraint.

2.3. Problem Formulation for the β-Robust Situation with Multiple Scenarios

Through solving the model (P_s), we can obtain the optimal solution for the p-median facility problem under any given scenario s, and calculate the corresponding optimal function value, z_s^*. Given a feasible solution for all scenarios, noted as X, the β-robust solution can be defined below.

Definition 1. *Solution X is β-robust for all scenarios when the following constraints are satisfied:*

$$\frac{z_s(X) - z_s^*}{z_s^*} \leq \beta, \ \forall s \in S, \tag{6}$$

where $z_s(X)$ represents the objective value for solution X under scenario s.

Then, the β-robust p-median facility location problem with uncertain demand can be formulated as follows.

$$\beta\text{-}P: \ Minz = \sum_{s \in S} \sum_{i \in M} \sum_{j \in N} q_s \sigma_{is} c_{ij} X_{ijs} \tag{7}$$

Subject to

$$\sum_{i \in M} \sum_{j \in N} \sigma_{is} c_{ij} X_{ijs} \leq (1 + \beta) z_s^*, \ \forall s \in S, \tag{8}$$

$$\sum_{j \in N} Y_j = p, \tag{9}$$

$$X_{ijs} \leq Y_j, \ \forall i \in M, \ j \in N, \ s \in S, \tag{10}$$

$$\sum_{j \in N} X_{ijs} = 1, \ \forall i \in M, \ s \in S, \tag{11}$$

$$X_{ijs}, \ Y_{js} \in \{0, 1\}, \ \forall i \in M, \ j \in N, \ s \in S. \tag{12}$$

Objective function (7) is to minimize the expected overall transportation cost. Submitting function (1) into (6), we can obtain Constraint (8), which ensures that the relative regret for any scenario $s \in S$ is no more than the required level β. The meaning of Constraints (9)–(12) is similar to the ones in P_s, while all these constraints must be satisfied for all scenarios $s \in S$.

3. Multi-Scenario Cooperative Evolutionary Algorithm

From the above model methodology, we can see that all of the subproblems P_s ($s \in S$) should be solved first, then the problem β-P can be solved based on the optimal solutions of P_s. Subproblem P_s is a typical p-median problem, which is usually solved by approximate metaheuristics [1]. When the number of scenarios is huge, it becomes quite time-consuming to obtain all of their (near) optimal solutions by repeatedly running a metaheuristic in the current literature. Thus, a multi-scenario cooperative evolutionary (MSCE) algorithm was developed to calculate the solutions for all scenarios in one run. The basic idea is that the optimal solutions to a set of (P_s) subproblems have higher similarities if the corresponding scenarios are similar to each other. The scenarios with higher similarities are called neighborhood scenarios. In the MSCE algorithm, the search processes for solving all (P_s) subproblems are parallelly conducted, and the knowledge of the solutions for similar scenarios is extracted and exchanged among these scenarios for a very certain number of iterations. The utilization of knowledge from the solutions of similar scenarios accelerates the convergence of the algorithm and helps find better solutions.

3.1. Solution Encoding

The solution of the p-median problem is encoded as a set with p elements, where each element represents the index of a selected facility for opening. For example, in a 5-median problem with 15

potential facility locations, {2,5,6,12,15} is a code that represents a feasible solution, and the potential locations, 2,5,6,12, and 15, are selected to establish facilities. This encoding strategy has been widely used in GA based algorithms [4,9], and can ensure that Constraint (2)/(9) is automatically satisfied.

3.2. Greedy Heuristics for Generating Initial Solutions

In this section, two greedy heuristics are presented to help generate near-optimal initial solutions for the subproblems under each scenario and the β-robust problem.

3.2.1. Greedy Adding Heuristic

The main idea of the greedy adding heuristic (GAH) is to find the potential facility locations sequentially with an attempt that can decrease the most value of the objective function (1), and add them into a set of open facilities one by one until the number of open facilities reaches p. The idea of the GAH has been utilized to solve different facility location problems [22,23].

The main procedure of the GAH is as follows.

Input: the parameters in model P_s;
the stopping criteria;
Output: the solution of P_s.
Step 0: ***Initialize***
let Q_o be the set of opened facilities and initialize $Q_o = \varnothing$;
let Q_u be the set of unopened facilities and initialize $Q_u = N$.
Step 1: ***calculate the objective value***
For $j \in Q_u$ **Do**
let $Q_{oj} = Q_o \backslash j$ and calculate function (1) to obtain z_{sj} for Q_{oj};
End for
Step 2: ***find and add***
find $j^* = \text{argmin}\{z_{sj}, j \in Q_u\}$;
add j^* into Q_o, and $Q_o = Q_o \cup j^*$;
remove j^* from Q_u, and $Q_u = Q_u \backslash j^*$;
Step 3: ***Check stopping criteria***
If the number of elements in $Q_o < p$ **Do**
go to Step 1;
Else
stop the algorithm and output the solution;
End if

3.2.2. Greedy Deleting Heuristic

The greedy deleting heuristic (GDH) can be viewed as a reverse process of GAH. First, all potential facilities in set N are assumed to be opened. Then, the facilities are closed one by one until the number of opened facilities is equal to p. In each process of closing one facility, the one whose closing can make the minimum objective value is found and deleted from the set of opened facilities. This greedy deleting strategy was also utilized by Berman et al. [24] and Alp et al. [4] for solving facility location problems. When the GAH is utilized to solve the β-P problem, only the one satisfying Constraint (8) can be deleted in each deleting operation.

The main procedure of the GDH is as follows.

Input: the parameters in model β-P;
the stopping criteria;
Output: the solution of β-P.
Step 0: ***Initialize***
let R_o be the set of opened facilities and initialize $R_o = N$;
let R_u be the set of unopened facilities and initialize $R_u = \varnothing$.
Step 1: ***calculate the feasible objective value***
For $j \in R_o$ **Do**
Let $R_{oj} = R_o \backslash j$;
calculate the function (1) for solution R_{oj} and check Constraint (8);
If solution R_{oj} can satisfy Constraints (8) for all scenarios **Do**
calculate function (7) for solution R_{oj} and note as z_j;
End if
End for
Step 2: ***find and delete***
find $j^* = \text{argmin}\{z_j\}$;
delete j^* from R_o, and $R_o = R_o \backslash j^*$;
Step 3: ***Check stopping criteria***
If the number of elements in $R_o > p$ **Do**
go to Step 1;
Else
stop the algorithm and output the solution;
End if

3.3. Knowledge Exchange with Neighborhood Scenario

The neighborhood relations between the scenarios are defined based on the distances between their demand vectors.

Definition 2. *The distance between two scenarios s and s′ is defined as follows:*

$$D_{ss'} = \sqrt{\sum_{i \in M} (\sigma_{is} - \sigma_{is'})^2} \tag{13}$$

where s and s′ ∈ S are the index of scenarios.

It can be seen that the smaller the value of $D_{ss'}$, scenario s is more similar to scenario s'. The optimal solution of the subproblem under scenario s should be similar to that of the subproblem under scenario s' if scenarios s and s' are similar to each other. Thus, the knowledge obtained from the solution of subproblem s should be helpful in optimizing subproblem s'. Based on this observation, three neighborhood knowledge utilizing (NKU) operators are employed to utilize the helpful knowledge in the solutions of the neighborhood scenarios.

(1) NKU operator based on random exchange

Let Y^s be the set of opened facilities in a local optimal solution found in the search process of solving subproblem corresponding scenario s and $Y^{s'}$ be that of scenario s'. Let Y^C be the common facilities appearing in Y^s and $Y^{s'}$, that is $Y^C = Y^s \cap Y^{s'}$. As there are similarities in the optimal solutions of scenario s and its neighborhood scenarios s', the common part Y^C is thought to be useful and is kept in the new solution. The other part of the solution $Y^{s'}$, that is $Y^{s'}/Y^C$, may also be part of the optimal solution of scenario s. Thus, some of the uncommon parts in Y^s and $Y^{s'}$ are randomly exchanged to form new solutions. To illustrate, we considered an instance with 15 potential facilities and $p = 5$. Let $Y^s = \{1, 3, 7, 10, 12\}$ be a local optimal solution of scenario s and $Y^{s'} = \{1, 3, 5, 8, 14\}$ be a local optimal solution of its neighborhood scenario s'. Then $Y^C = \{1, 3\}$, $Y^s/Y^C = \{7, 10, 12\}$, and $Y^{s'}/Y^C = \{5, 8, 14\}$. Some of the uncommon parts in Y^s and $Y^{s'}$ are randomly selected (e.g., 10 in Y^s/Y^C and 8 in $Y^{s'}/Y^C$)

and exchanged to form two new solutions, {1, 3, 7, 8, 12} and {1, 3, 5, 10, 14}. Finally, the better one of the two new solutions is selected.

(2) NKU operator based on the GAH

In the NKU operator based on the GAH, the common parts of the solutions for scenario s and its neighborhood scenario s' are also kept in the new solution. The number of facilities in Y^C is usually less than p. In this situation, the number of $p - \left|Y^C\right|$ facilities are selected by the GAH from all of the unopened facilities (N/Y^C) to form a new solution.

(3) NKU operator based on the GDH

In the NKU operator based on the GDH, the opened facilities in Y^s and $Y^{s'}$ are united to form a set $Y^U = Y^s \cup Y^{s'}$. Then, the facilities in Y^U are deleted one by one using GDH until the number of facilities is equal to p.

The above three NKU operators can exchange some useful knowledge between the solutions of a scenario and its neighborhood. In the MSCE algorithm, these NKU operators are randomly employed.

3.4. Local Search

After the knowledge is exchanged between the solutions of a scenario and its neighborhood, a new solution is generated for the scenario. Then, using this new solution as a starting node, some local searches are conducted to improve the solution. Two local search operators are employed in the local search.

(1) Random exchange

In the random exchange local search operator, a random number of opened facilities in the current solution is randomly selected and exchanged with the same number of facilities randomly selected from the unopened facilities.

(2) Neighborhood exchange

The neighbor exchange local search operator is designed based on a definition of neighborhood facility, which is introduced as follows.

Definition 3. *The kth neighborhood facility of facility j is the facility that is the kth closest to facility j among all of the other potential facilities.*

In this operator, the randomly selected facilities only exchange open decisions with their neighborhood facilities. For example, {2, 5, 7, 10, 12} are the first five closest facilities of 3. If facility 3 in the solution is selected and exchanged with an unopened facility, one of the unopened facilities in set {2, 5, 7, 10, 12} will be selected for exchange with 3.

3.5. Framework of MSCE

For conducting the MSCE algorithm, the neighborhood scenarios/facilities for each scenario/facility are first found in the initialization step. Then, the GAH algorithm is employed to calculate a local optimal solution for all scenarios, which serve as the starting search point. The GDH algorithm is used to calculate a local optimal solution for the β-P problem, as it has more opportunities to generate a feasible solution that satisfies Constraint (8). In the second step, the useful knowledge on solving a scenario subproblem is extracted and utilized to generate a new solution by randomly employing one of the three NKU operators. In the third step, a number of local search operations are conducted from the new solution obtained in Step 2. The local search process is parallelly conducted and the number of iterations is set as 100. In the fourth step, the local optimal solution obtained in Step 3 is checked to see if it can improve the objective of its neighborhood scenarios. The last step is to check if the stopping criteria are satisfied.

The main procedure of the MSCE algorithm is presented as follows:

Input: the parameters in model P_s and β-P;
the stopping criteria;
Ns, the number of neighborhood scenarios.
Output: the optimal solutions for all P_s ($s \in S$) and β-P.
Step 0: ***Initialize***
calculate the distances between each two scenarios and obtain their neighborhoods;
calculate the distances between each two facilities and obtain their neighborhoods.
Step 1: ***Generate initial solutions by greedy heuristics***
For i = 1: $|S|$ **Do**
calculate the solution for subproblem s by GAH heuristic;
End for
calculate the solution for β-P by GDH heuristic.
Step 2: ***Exchange knowledge between neighborhood scenarios***
For s = 1: $|S|$ **Do**
randomly select a NKU operator to generate a new solution;
End for
generate a new solution for β-P by the NKU operator based on GDH;
Step 3: ***Parallel local search***
Parfor all P_s ($s \in S$) and β-P **Do**
For i = 1: T **Do**
generate a new solution by a randomly selected local search operator;
update the solution if the newly generated one is better;
End for
End parfor
Step 4: ***Update solutions for neighborhood scenarios***
For all P_s ($s \in S$) and β-P **Do**
For i = 1: Ns **Do**
update neighbor i's solution if the solution of scenario s is better;
End for
End for
Step 5: ***Check stopping criteria***
If one of the stopping criteria is satisfied **Do**
stop the algorithm and output the solutions;
Else
go to Step 2;
End if

In this paper, the number of neighborhood scenarios, Ns, was set as 20. The number of local search iterations, T, was set as 100. Two stopping criteria were applied: (1) the number of iterations conducting Step 2 was over 1000, and (2) if there was no improvement in the solution for 10 iterations conducting Step 2.

4. Computational Experiments

In this section, the performance of the proposed algorithm was analyzed based on the computational experiments for two practical cases. All of the experiments are conducted on an Intel(R) Core i7, 2.70 GHz, CPU 16GB of a RAM computer with Windows 10 system. All algorithms were implemented by MATLAB (2016a, MathWorks, Natick, MA, USA).

4.1. Illustration of Two Practical Cases

Two typical networks for facility location were selected to test the proposed algorithm. The first was a case with 95 cities distributed in Hunan Province in China, noted as H-95, and the distribution of the cities is presented in Figure 1a. Each city represents both a demand node and a potential location for opening a new facility. The distance between each two nodes was assumed to be the greatest circle

distance, which was calculated through their practical longitudes and latitudes. The excepted demand of each customer node was estimated through the population in the city. In each scenario, the realizing demand of each node was assumed to have three possible levels, which were the excepted demand, 25% higher, or lower than the excepted demand. In the experiment of the 95-node instance, 100 possible scenarios for the demands were generated by randomly selecting a demand level for each node.

(**a**) (**b**)

Figure 1. Node distribution for the two practical cases. (**a**) Ninety-five cities in Hunan Province and (**b**) 668 demand points in Changsha city.

The second case was based on 668 demand nodes distributed in the urban area of Changsha in China, noted as C-668, which are presented in Figure 1b. Among the 668 nodes, 150 nodes were selected as the potential locations for opening new facilities. The distance between each pair of two nodes was estimated by the Manhattan distance based on their practical longitudes and latitudes. The excepted demand in each node was estimated by the population surrounding the demand node. In this case, the realizing demand of each node in a scenario was assumed to have five possible levels, which were 50%, 75%, 100%, 125%, and 150% of the excepted demand. Additionally, 100 possible demand scenarios were generated through randomly selecting a demand level for each node.

In both the H-95 and C-668 cases, the probability that each scenario may occur was generated randomly, and $\sum_{s \in S} q_s = 1$. The cost c_{ij} was set as the distance between node i and j.

4.2. Experimental Results and Analysis

In the experiment, the value of p was set as 10, 20, and 30, respectively for both the H-95 and C-668 cases. Thus, we had six instances with different sizes, and all were solved by the proposed MSCE algorithm 51 times. The parameters of the MSCE algorithm were set as per the illustration in Section 3.5. To illustrate the performance of the MSCE algorithm, a location and allocation heuristic (LAH) was also used to solve all instances. The LAH algorithm is a commonly used approach for solving different facility location problems including the p-median problem, and can provide relatively good solutions [23,25]. More details of the LAH can be found in Jia et al. [23]. The average results by the MSCE for 51 runs and the results by the LAH for all instances are presented in Figure 2. We can see that the MSCE algorithm outperformed the LAH algorithm for all scenarios in the six different instances.

The statistical results for the computational time of the MSCE algorithm in all instances is reported in Table 1. It can be seen that the computational time increased as the size of the instance became bigger. Considering there were 100 scenarios in each instance, the computational time was acceptable for solving the problem.

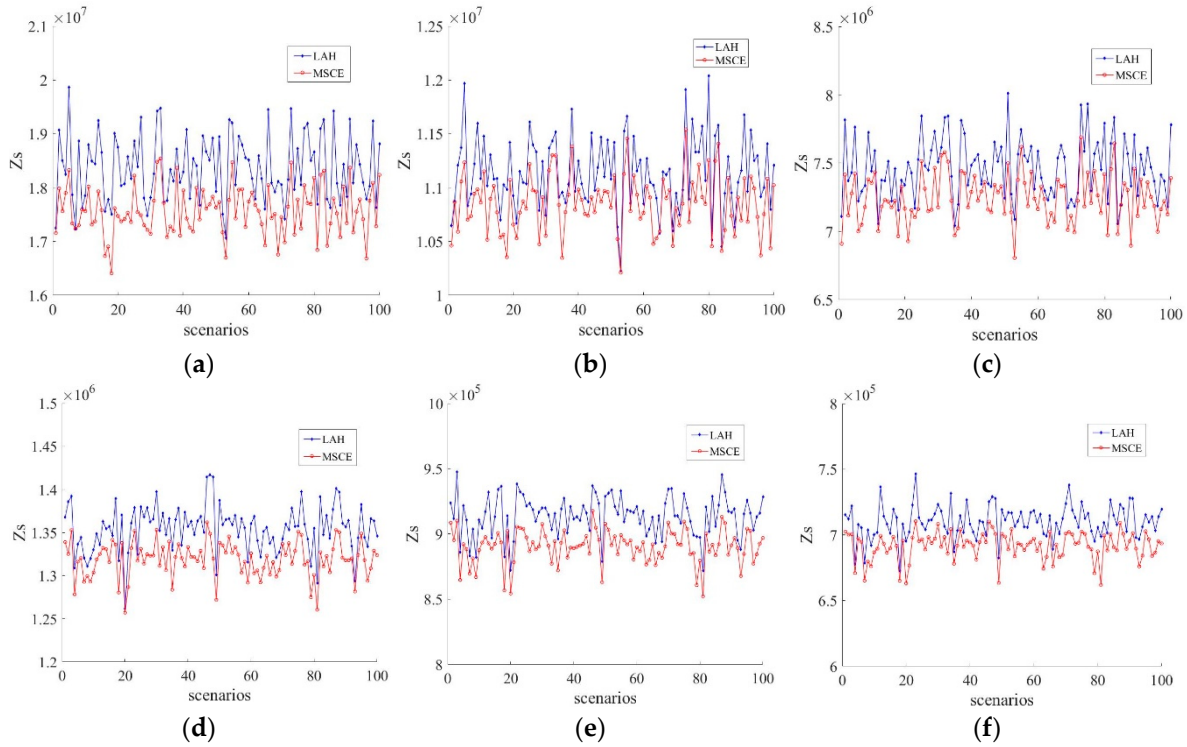

Figure 2. The computational results for H-95 and C-668. (**a**) H-95 with $p = 10$; (**b**) H-95 with $p = 20$; (**c**) H-95 with $p = 30$; (**d**) C-668 with $p = 10$; (**e**) C-668 with $p = 20$; and (**f**) C-668 with $p = 30$.

Table 1. The computational time of 51 runs for all 100-scenario instances.

Cases	M/N	p	Computational Time (Seconds)	
			Average	Standard Deviation
H-95	95/95	10	122.861	37.353
		20	318.011	66.745
		30	642.455	164.929
C-668	668/150	10	531.771	115.468
		20	1211.889	370.137
		30	1985.167	452.943

The GA is also widely used to solve p-median problems. Here, we utilized the standard framework of the GA and the general operators used in Medaglia et al. [26] to solve the problem. No enforcements on the initial population and genetic operators were considered as these would usually further increase the computational time. The population size of GA was 100 and the maximum number of evolution generations was 1000. The GA will also stop if there is no improvement in the solution for 100 generations. The mutation probability was 0.1. The GA was run 51 times for each scenario individually. When we solved the instances for H-95 and C-668 with $p = 10$, we found that the statistical results on the total computational time by GA for all 100 scenarios (as presented in Figure 3) were relatively large. The experiments for the other larger size instances became unacceptable with time. Thus, we report the results for these two instances here. The solution obtained by the GA was very close to that by the MSCE algorithm. We present the relative value of the GA to the MSCE in Figure 4. As shown in Figure 4a, among the results of 100 scenarios for the H-95 case with $p = 10$, where 15% obtained by the GA was better than that of the MSCE; 7% obtained by the GA was worse than that of the MSCE; and the other 78% were equal to each other. Among the results of the 100 scenarios for the C-668 case with $p = 10$, only 5% obtained by GA was better than that of MSCE, while 93% obtained by the MSCE was better than that of the GA. Thus, the MSCE performed much better than the GA in the C-668 case.

The main reason may be that the neighborhood scenarios in the C-668 case had higher similarities, and more helpful knowledge could be utilized by the MSCE algorithm. In all, compared with the GA, the MSCE algorithm could obtain similar or better results, while the computational time of the MSCE was much shorter than that of the GA.

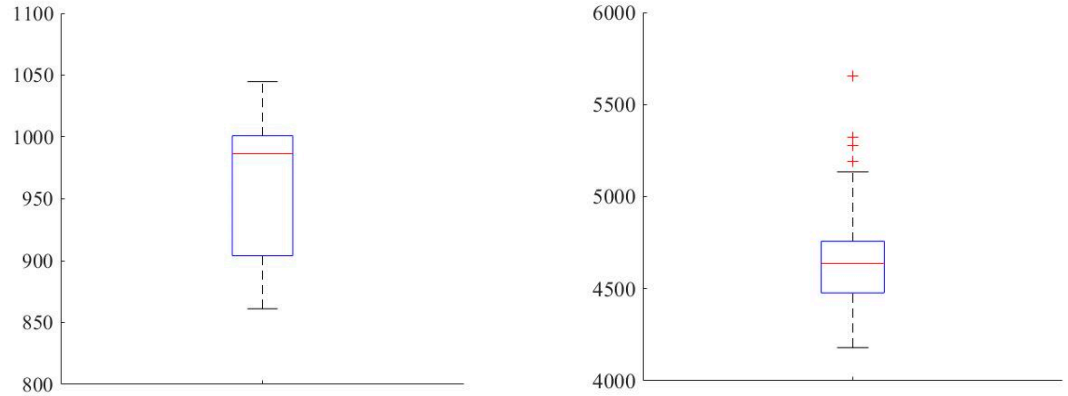

Figure 3. The statistical results of the total computational time for all 100 scenarios by the genetic algorithm (GA). (**a**) H-95 with $p = 10$; (**b**) C-668 with $p = 10$.

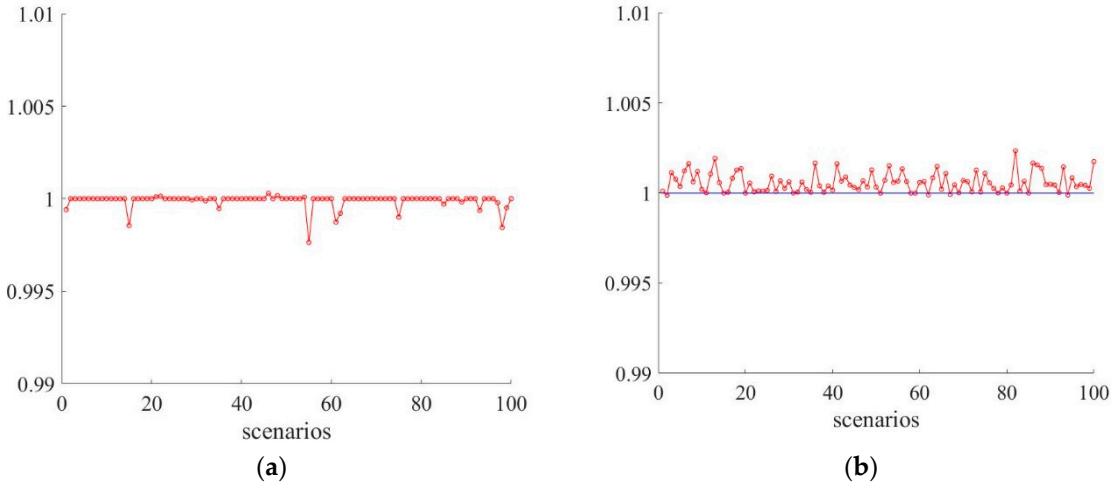

Figure 4. The relative objective value obtained by the GA compared to that of the multi-scenario cooperative evolutionary (MSCE). (**a**) H-95 with $p = 10$; (**b**) C-668 with $p = 10$.

5. Discussion and Conclusions

The main motivation behind the use of the MSCE algorithm was to sufficiently utilize the knowledge obtained from the solutions of neighborhood scenarios to improve the overall search efficiency of solving all scenarios. The computational results for the two practical cases (H-95 and C-668) with 100 scenarios confirmed that the proposed algorithm could obtain better solutions in a much shorter time when compared to the LAH algorithm and GA. The algorithm was suitable for solving problems with a large number of scenarios, as more similar scenarios would be found in a large set of scenarios. If there are a small number of scenarios that are quite different to each other in the problem, there would not be many similarities between these optimal solutions, that is, the MSCE algorithm cannot find sufficient useful knowledge. In this situation, the traditional approximate metaheuristics may be more suitable.

In this work, six instances with different sizes generated from two practical cases were tested and compared in the experiment. In future research, more experiments for different cases should be conducted to further test the performance of the proposed algorithm. The proposed algorithm can also be extended to solve many other optimization problems under uncertain scenarios, for example, the path planning problem in [27], the portfolio optimization problem in [28], and the vehicle routing problem with uncertain demand [29]. Aside from the LAH and GA, other metaheuristics can be compared with the MSCE algorithm. Three neighborhood knowledge utilizing operators are designed to draw and use the helpful knowledge in solutions of neighborhood scenarios. It would be a valuable direction to study new learning strategies for utilizing the similarities among scenarios, then, a more efficient algorithm can be developed.

Author Contributions: Methodology, J.S. and R.W.; Validation, X.Z.; Formal analysis, J.S. and X.Z.; Investigation, B.J.; Data curation, J.S.; Writing—original draft preparation, J.S.; Writing—review and editing, X.Z.; Visualization, B.J.; Supervision, J.S.; Funding Acquisition, J.S.

Funding: This research was supported by the National Natural Science Foundation of China (grant nos. 71771215, 61773390), the Natural Science Fund of Distinguished Young Scholars in Hunan (grant no. 2018JJ1035), the key project of National University of Defense Technology (ZK18-02-09), and a project from the Department of Education of Guangdong Province (Grant no. 2018KTSCX245).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Mladenović, N.; Brimberg, J.; Hansen, P.; Moreno-Pérez, J. The p-median problem: A survey of metaheuristic approaches. *Eur. J. Oper. Res.* **2007**, *179*, 927–939. [CrossRef]
2. Cornuejols, G.; Fisher, M.L.; Nemhauser, G.L. Location of Bank Accounts to Optimise Float: An Analytic Study of Exact and Approximate Algorithms. *Manag. Sci.* **1977**, *23*, 789–810. [CrossRef]
3. Hosage, C.M.; Goodchild, M.F. Discrete space location-allocation solutions from genetic algorithms. *Ann. Oper. Res.* **1986**, *6*, 35–46. [CrossRef]
4. Alp, O.; Erkut, E.; Drezner, Z. An efficient genetic algorithm for the p-median problem. *Ann. Oper. Res.* **2003**, *122*, 21–42. [CrossRef]
5. Chiyoshi, F.; Galvão, R.D. A Statistical Analysis of Simulated Annealing Applied to the *p*-Median Problem. *Ann. Oper. Res.* **2000**, *96*, 61–74. [CrossRef]
6. Correa, E.S.; Steiner, M.T.A.; Freitas, A.A.; Carnieri, C. A Genetic Algorithm for solving a capacitated *p*-median Problem. *Numer. Algorithms* **2004**, *35*, 373–388. [CrossRef]
7. Alba, E.; Dominguez, E. Comparative analysis of modern optimization tools for the *p*-median problem. *Stat. Comput.* **2006**, *16*, 251–260. [CrossRef]
8. Kratica, J.; Stanimirović, Z.; Tošić, D.; Filipović, V. Two genetic algorithms for solving the uncapacitated single allocation p-hub median problem. *Eur. J. Oper. Res.* **2007**, *182*, 15–28. [CrossRef]
9. Li, X.; Xiao, N.; Claramunt, C.; Lin, H. Initialization strategies to enhancing the performance of genetic algorithms for the *p*-median problem. *Comput. Ind. Eng.* **2011**, *61*, 1024–1034. [CrossRef]
10. Herda, M. Parallel genetic algorithm for capacitated *p*-median problem. *Procedia Eng.* **2017**, *192*, 313–317. [CrossRef]
11. Memari, A.; Ahmad, R.; Jokar, M.R.A.; Rahim, A.R.A. A New Modified Firefly Algorithm for Optimizing a Supply Chain Network Problem. *Appl. Sci.* **2019**, *9*, 7. [CrossRef]
12. Gola, A.; Klosowski, G. Development of computer-controlled material handling model by means of fuzzy logic and genetic algorithms. *Neurocomputing* **2019**, *338*, 381–392. [CrossRef]
13. Burduk, A.; Musial, K. Genetic algorithm adoption to transport task optimization. *Adv. Intell. Syst. Comput.* **2016**, *527*, 366–375.
14. Gola, A.; Klosowski, G. Application of fuzzy logic and genetic algorithms in automated works transport organization. *Adv. Intell. Syst. Comput.* **2018**, *620*, 29–36.
15. Snyder, L.V. Facility location under uncertainty: A review. *IIE Trans.* **2006**, *38*, 537–554. [CrossRef]

16. Weaver, J.R.; Church, R.L. Computational procedures for location problems on stochastic networks. *Transp. Sci.* **1983**, *17*, 168–180. [CrossRef]
17. Chen, G.; Daskin, M.S.; Shen, Z.M.; Uryasev, S. The a-reliable mean-excess regret model for stochastic facility location modeling. *Nav. Res. Logist.* **2006**, *53*, 617–626. [CrossRef]
18. Snyder, L.V.; Daskin, M.S. Stochastic *p*-robust location problems. *IIE Trans.* **2006**, *38*, 971–985. [CrossRef]
19. Lim, G.; Sonmez, A. γ-Robust facility relocation problem. *Eur. J. Oper. Res.* **2013**, *229*, 67–74. [CrossRef]
20. Sakalli, U.S.; Ataba, I. Ant Colony Optimization and Genetic Algorithm for Fuzzy Stochastic Production-Distribution Planning. *Appl. Sci.* **2018**, *8*, 2042. [CrossRef]
21. Shi, J.; Zhang, G.; Sha, J. A Lagrangian based solution algorithm for a build-to-order supply chain network design problem. *Adv. Eng. Softw.* **2012**, *49*, 21–28. [CrossRef]
22. Jain, K.; Mahdian, M.; Saberi, A. A new greedy approach for facility location problems. In Proceedings of the 34th Annual ACM Symposium on Theory of Computing, Montreal, QC, Canada, 19–21 May 2002; pp. 731–740.
23. Jia, H.; Ordóňez, F.; Dessouky, M. Solution approaches for facility location of medical supplies for large-scale emergencies. *Comput. Ind. Eng.* **2007**, *52*, 257–276. [CrossRef]
24. Berman, O.; Drezner, Z.; Wesolowsky, G. Locating service facilities whose reliability is distance dependent. *Comput. Oper. Res.* **2003**, *30*, 1683–1695. [CrossRef]
25. Taillard, E.D. Heuristic methods for large centroid clustering problems. *J. Heuristics* **2003**, *9*, 51–73. [CrossRef]
26. Medaglia, A.L.; Villegas, J.G.; Rodríguez-Coca, D.M. Hybrid biobjective evolutionary algorithms for the design of a hospital waste management network. *J. Heuristics* **2009**, *15*, 153–176. [CrossRef]
27. Zhao, T.; Huang, J.; Shi, J.; Chen, C. Route Planning for Military Ground Vehicles in Road Networks under Uncertain Battlefield Environment. *J. Adv. Transp.* **2018**, *2018*, 2865149. [CrossRef]
28. Zhou, X.; Wang, H.; Peng, W.; Ding, B.; Wang, R. Solving multi-scenario cardinality constrained optimization problems via multi-objective evolutionary algorithms. *Sci. China Inf. Sci.* **2019**, *62*, 1–18. [CrossRef]
29. Sungur, I.; Ordónez, F.; Dessouky, M. A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty. *IIE Trans.* **2008**, *40*, 509–523. [CrossRef]