

## Research Article

# Location of Facility Based on Simulated Annealing and “ZKW” Algorithms

Yukun Dong,<sup>1</sup> Jinyu Wang,<sup>1</sup> Fenghua Chen,<sup>1</sup> Yong Hu,<sup>2</sup> and Yong Deng<sup>2,3,4</sup>

<sup>1</sup>College of Information Science and Technology, Jinan University, Tianhe, Guangzhou 510632, China

<sup>2</sup>Big Data Decision Institute, Jinan University, Tianhe, Guangzhou 510632, China

<sup>3</sup>Institute of Fundamental and Frontier Science, University of Electronic Science and Technology of China, Chengdu 610054, China

<sup>4</sup>School of Computer and Information Science, Southwest University, Chongqing 400715, China

Correspondence should be addressed to Yong Deng; [ydeng@jnu.edu.cn](mailto:ydeng@jnu.edu.cn)

Received 13 June 2017; Revised 8 August 2017; Accepted 14 August 2017; Published 28 September 2017

Academic Editor: Erik Cuevas

Copyright © 2017 Yukun Dong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To cope with the facility location problem, a method based on simulated annealing and “ZKW” algorithm is proposed in this article. The method is applied to some real cases, which aims to deploy video content server at appropriate nodes in an undirected graph to satisfy the requirements of the consumption nodes with the least cost. Simulated annealing can easily find the optimum with less reliance on the initial solution. “ZKW” algorithm can find the shortest path and calculate the least cost from the server node to consumption node quickly. The results of three kinds of cases illustrate the efficiency of our method, which can obtain the optimum within 90 s. A comparison with Dijkstra and Floyd algorithms shows that, by using “ZKW” algorithm, the method can have large iteration with limited time. Therefore, the proposed method is able to solve this video content server location problem.

## 1. Introduction

Facility location problem, first introduced by Alfred and Friedrich [1], is a classical and well-studied problem. The goal of this problem is to find optimal locations to build facilities so that the solution of the problem can serve the consumers or clients with the least cost. Generally, the facility location problem could be divided into median problem [2, 3], covering problem [4–7], center problem [8–10], multiproduct problem [11, 12], dynamic location problem [13, 14], multiobjective location problem [15–17], competitive problem [18, 19], network reliability problem [20, 21], and network center location problem [22, 23].

All these branches are applied to many fields like fire-fighting units [24], emergency services [16, 25, 26], healthcare location [27], gas marketing hubs [28], factory sites [29], supply chain network [30–32], and so on [33]. Nowadays, with the development of Internet, more and more people surf the Internet to watch videos. And the fluency and quality of a video are vital because they will have a great impact on people’s viewing experience. Although it is quite important to determine how to deploy the server to meet the requirements

of consumers for a video service provider, the practical application of the problem is limited.

In this article, a method based on the simulated annealing and “ZKW” algorithms is proposed to deal with an application in such area. Simulated annealing algorithm, developed by Kirkpatrick et al. [34, 35], can efficiently solve the NP problem with the advantages of avoiding falling into local optimum and less reliance on the initial solution; therefore, it is widely applied in many areas, such as image processing [36, 37], vehicle routing [38, 39], production scheduling [40], and machine learning [41]. “ZKW” algorithm, proposed by ZKW [42], is an efficient and fast minimum-cost flow algorithm. We try to use simulated annealing algorithm to search optimum, which is searching some network nodes in the undirected graph to deploy video content server. At the same time, “ZKW” algorithm is used to find the shortest path and calculate the least cost from the video content server to the consumption node. The result of cases given by HUAWEI Code Craft illustrates that the proposed method can efficiently solve the facility location problem and find a good plan with limited time. What is more, a comparison with

traditional shortest path algorithms, Dijkstra [43] and Floyd [44] algorithms, shows that, by using “ZKW” algorithm, the running speed is greatly improved.

The paper is organized as follows. Section 2 introduces some basic concepts about the simulated annealing and “ZKW” algorithm. Section 3 is an application in the cases given by HUAWEI Code Craft, which describes the problem and illustrates the specific model of simulated annealing and “ZKW” algorithm, and gives a pseudocode. The result analyses are given in Section 4 to indicate the effectiveness of the method. The conclusions are drawn in Section 5.

## 2. Preliminaries

In this section, some basic concepts are briefly introduced.

**2.1. Simulated Annealing.** Simulated annealing (SA), first proposed by Metropolis et al. [45] in 1953 and further developed by Kirkpatrick et al. [34, 35] in 1983, is a stochastic searching optimization algorithm. The algorithm is deprived of the solid-annealing principle. It refers to heating the solid material at an enough high temperature, at which the inner particles of the material will become disordered and the internal energy will increase. Then, bringing down the temperature to decrease defect and reach an equilibrium at each temperature, the system’s energy is minimized [35].

In optimization problem, the simulated annealing algorithm is explained as lowering the temperature of the system until it converges to a feasible and steady solution. During the process, while the temperature is high, accepting a solution that is worse than the current solution will be allowed to ensure the algorithm to jump out of local optimum. In the meantime, a slow decrease in the probability of accepting a worse solution with exploring the solution space is adopted to allow the algorithm to gradually focus on an area to search the optimum solution [35].

Suppose that there is an optimization problem, which is

$$\min f(x), \quad x \in S, \quad (1)$$

where  $f(x)$  is an objective function,  $S$  is a finite solution space, and  $x$  is the current solution. The solution steps of simulated annealing algorithm (the flow chart is shown in Figure 1) could be depicted as follows:

- (1) Choose an initial solution  $x \in S$  at random and select an initial temperature  $T_0$  as well as a final temperature  $T_f$ .
- (2) Generate a domain solution  $y \in N(x)$ , where  $N(x)$  is the domain structure of  $x$ ; calculate the increment of the objection function  $\Delta f = f(y) - f(x)$ .
- (3) If  $\Delta f < 0$ , then accept the new solution,  $x = y$ ,  $f(x) = f(y)$ , and turn to the next step. Otherwise, generate a random number  $\zeta = U(0, 1)$ . According to the Metropolis criterion, if  $\exp(-\Delta f/T_k) > \zeta$ , accept the new solution,  $x = y$ ,  $f(x) = f(y)$ .
- (4) If the heat balance (the inner cycle number larger than  $n(T_k)$ ) is reached, turn to the next step. Otherwise, go back to the second step.

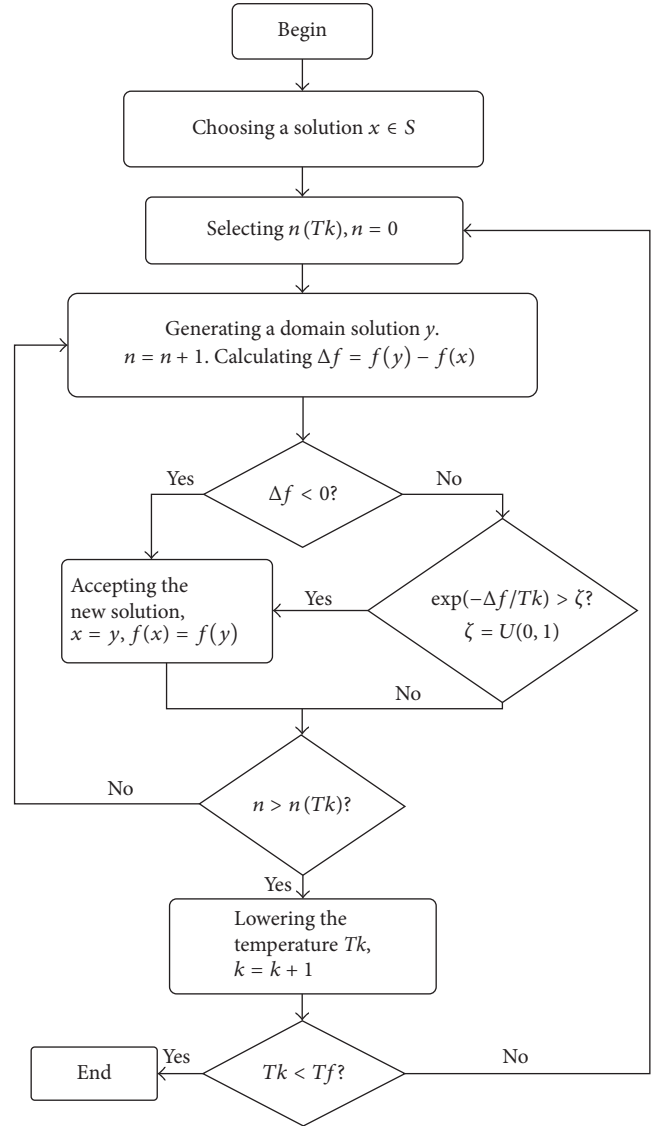


FIGURE 1: The flow chart of simulated annealing algorithm.

- (5) Cool down the temperature  $T_k$ , and  $k = k + 1$ . If  $T_k < T_f$ , stop the algorithm. Otherwise, go back to the second step.

**2.2. “ZKW” Algorithm.** “ZKW” algorithm, proposed by ZKW [42], is a minimum-cost flow algorithm. The major idea of the algorithm is finding the augmenting path. Before further discussing the algorithm, the property of the shortest path will be introduced. Suppose that  $dis[i]$  is the shortest distance from node  $i$  to the starting node and that  $w(i, j)$  is the distance between node  $i$  and node  $j$ :

- (1) For any edge  $(i, j)$  in the shortest path, it will satisfy  $dis[i] \leq dis[j] + w(i, j)$ .
- (2) For any node  $i$  in the shortest path, there will be a node  $j$  that will satisfy  $dis[i] = dis[j] + w(i, j)$ .

In the process of finding a shortest path, if we follow the rule of  $dis[i] \leq dis[j] + w(i, j)$  to augment, it will satisfy

the first property. However, the second property may not be satisfied. In order to find a new augmenting path, the Dijkstra or Floyd algorithm is needed to recalculate and find out node  $j$  that satisfies the second property, which will waste lots of time. Thus, we can use the strategy of the KM algorithm [46–48]. Every time, the edge that satisfies the second property will be augmented. If the path cannot reach the ending node,  $\text{dis}[j]$  of node  $j$  that is on the augmenting path (ap) will add  $\Delta$ , where  $\Delta = \min_{j \in \text{ap}, i \notin \text{ap}} (\text{dis}[i] + w(i, j) - \text{dis}[j])$ . Therefore, there will be another edge that satisfies the second property that can be added.

The steps of the algorithm are as follows:

- (1) Clear the array  $\text{dis}$ , which means that there is only a starting node in the current subgraph.
- (2) For the depth-first search, if in this process the ending node can be reached, backtrack and minus the flow that has been used. Then, run this step again. If the ending node cannot be reached, turn to the next step.
- (3) Add  $\Delta$  to  $\text{dis}[j]$  and find another edge to add in the subgraph until no more edge can be added, thus, end.

### 3. Application in Facility Location Problem

In this section, the proposed method is described in detail when applying to real cases given by [49].

#### 3.1. Problem Description

*Network Structure Model* [49]. The real world is very complicated; complex network is used to model the uncertainty and complexity [50–54]. Facility location problem can be seen as a network optimization [55]. Many methods for optimization and decision-making under uncertainty, such as fuzzy sets [56–58], evidence theory [59], and D numbers [60–62], are applied to address this issue [63, 64]. For the sake of simplicity, the problem is abstracted into a given undirected graph that consists of some network nodes. Each node is connected with at least one node through the network link. At the same time, the node can transmit data through the network link to the other nodes that are connected. The total bandwidth and the rental fee of each network link are different. The network link needs to charge the rental fee based on the usage of bandwidth.

*Consumption Node* [49]. In the given undirected graph, some nodes are connected with the consumption nodes directly. The consumption requirement of each consumption node is different. One consumption node can obtain video data from several video content servers at the same time.

*Video Content Server* [49]. The video content server can provide infinite video data, which means that it can serve several consumption nodes. The data is transmitted to the consumption node through the network path that is composed of network nodes and network links.

*Requirements* [49]. Under the condition of satisfying the requirements of all consumption nodes, select some network

nodes from the given network structure model and deploy the video content server reasonably and properly to receive the least cost, which consists of the cost of deploying the video content servers and rental fee.

#### 3.2. Model of Simulated Annealing

*The Objective Function.* The problem requires choosing some appropriate network nodes to deploy the video content server so that the requirements of the consumption nodes could be satisfied with the least cost. The cost consists of the servers' cost and the rental fee. Therefore, we can obtain the objection function, which is

$$\min \text{Cost} = m \cdot \text{SC} + \sum_{i=1}^N \text{CC}_i, \quad (2)$$

where  $m$  is the total number of video content servers that need to be deployed; SC is the cost of video content server;  $N$  is the number of consumption nodes; thus, there are  $N$  network paths connecting the consumptions nodes and video content servers;  $\text{CC}_i$  is the cost of rental fee for  $i$ th consumption node.

*The Domain Structure.* Different domain structure will have a different influence on finding a good solution for the simulated annealing. In solving the problem, we propose three kinds of domain structures listed as follows:

- (i) Reducing a video content server at random, which means a network node that has been deployed with a video content server will be chosen randomly and removed from the current deploying plan
- (ii) Adding a video content server at random, which means a network node that has not been deployed with a video content server will be chosen randomly and added to the current deploying plan
- (iii) Changing a video content server at random, which means a network node that has been deployed with a video content server in the current deploying plan will be chosen randomly and replaced by another network node

When choosing the structure at each iteration, the random number is used. That is, generating a random number  $a = U(0, 1)$ ,

domain structure

$$= \begin{cases} \text{"Reducing"} \text{ structure,} & a \leq a_1 \\ \text{"Adding"} \text{ structure,} & a_1 < a \leq a_2 \\ \text{"Changing"} \text{ structure,} & a > a_2, \end{cases} \quad (3)$$

where  $a_1, a_2$  are the parameters used to adjust the selective probability of different domain structures.

*The Temperature Parameters.* Generally, there are two widely used cooling schedules, which are  $T_{t+1} = T_t \cdot \alpha$  and  $T_{t+1} = T_t - \Delta T$ , where  $\alpha$  and  $\Delta T$  are constants. The schedules are

simple and practicable. However, the temperature does not cool down quickly; thus, we will adopt another schedule proposed by Lundy and Mees [65], which is  $T_{t+1} = T_t / (1 + \beta \cdot T_t)$ , where  $\beta$  is a constant. When the temperature is high, the temperature of the schedule can cool down with a large interval. As the number of iterations increases, the schedule will have slower cooling than a proportional temperature reduction with fixed  $\alpha$  and  $\Delta T$ . Therefore, the algorithm can search the optimum solution on an area. The comparison of different cooling schedules will be discussed in the next section.

In order to avoid being trapped in a local optimum, we suggest returning to a higher temperature and search again when the temperature is cooled down at a relatively low level.

The initial temperature and the final temperature will also have a great influence on the performance of the simulated annealing. In general, the initial temperature should be larger so that the algorithm can be in equilibrium at the beginning. And the final temperature should be smaller so that the algorithm can have enough time to reach an optimal solution.

**3.3. Model of “ZKW” Algorithm.** In finding an optimum plan, we need to obtain the shortest path (the least cost) from the video content server to the consumption node. Generally, Dijkstra [43] and Floyd [44] are the two algorithms that are suited to the issue. However, the time complexity of Floyd algorithm is  $O(n^3)$ , and that of Dijkstra algorithm is  $O(n^2)$ . Neither is fast enough to get an optimal solution with limited time, which is 90 seconds in the problem. As being introduced above, “ZKW” algorithm can save the time that is used to recalculate  $dis[i]$  every time, which is huge waste. Thus, “ZKW” algorithm is much better for applying to the problem.

**3.4. The Pseudocode.** See Pseudocode 1.

## 4. Result Analyses

**4.1. Operating Environment.** The details of the operating environment are given as follows:

CPU: Intel(R) Core(TM) i5-3337U CPU @ 1.80 GHz  
 Memory: 4.00 GB  
 Kernel: Two cores  
 Compiler: Java 1.7.0\_71  
 Operation system: 64-bit Windows 10, professional edition

**4.2. The Result of the Proposed Method.** Three kinds of cases will be applied to test the proposed method. The first case is a primary case, which contains 160 network nodes and 609 network links in the undirected graph with 72 consumption nodes. The second case is an intermediate case, which contains 300 network nodes and 1169 network links in the undirected graph with 135 consumption nodes. And the third case is an advanced case, which contains 800 network nodes and 2959 network links in the undirected graph with

360 consumption nodes. All data come from the preliminary contest in [49]. The results are shown in Table 1.

As shown in Table 1, for primary case, the total iteration can reach over 170 thousand within 90 seconds, which means the algorithm can process a great amount of searching. Thus, it can be easy to find an optimum. Similarly, the algorithm can also reach high iterations and relatively stable solutions in the intermediate and advanced cases.

In general, the proposed method not only is fast enough but also is able to find an optimal solution. So it is suitable for this facility location problem that needs to deploy the video content sever at appropriate network node to satisfy the requirement of the consumption nodes in the undirected graph.

**4.3. Comparison of Different Methods to Calculate the Least Cost.** In this problem, the time is limited to 90 s; thus, the running speed is quite important: the faster the better. In searching appropriate network node to deploy the video content server in the undirected graph, the shortest path from the server to the consumption node needs to be found out and calculated. Generally, Dijkstra and Floyd algorithms are used to calculate the shortest path. However, the time complexities of algorithms are too high. “ZKW” algorithm provides us with another method to find the shortest path. “ZKW” is a cost flow algorithm. The problem can be explained as finding the least cost with set flow. And, by using the “ZKW” algorithm, the augmenting process is actually a process of finding the shortest path, so the time could be greatly saved. A comparison of Floyd, Dijkstra, and “ZKE” algorithms under the intermediate case is shown in Table 2.

It is easy to find in Table 2 that the iteration of “ZKW” algorithm incredibly increases when compared with Floyd and Dijkstra algorithms. In a condition that ensured the high speed, the method that uses “ZKW” algorithm can find a much better solution. By the way, although the iteration of “ZKW” algorithm is almost 100 times as large as that of Dijkstra algorithm, the least cost only reduces about 2000. It means that a large number of iterations failed or that they might be trapped in a local optimum. So the proposed method could be further improved.

**4.4. Determination of Temperature Constants  $\beta$  and  $\alpha$ .** In the article, the cooling schedule  $t = t / (1 + \beta t)$  is chosen instead of  $t = \alpha t$ . In order to illustrate the reason, the performances of these two schedules are discussed. And the most reasonable  $\beta$  and  $\alpha$  need to be found out firstly. Suggested by [65],  $\beta$  should be less than  $1/U$ , where  $U$  is any upper bound on  $\max \Delta f$ . In the advanced case,  $U$  is about 1000. And, typically, the value of  $\alpha$  lies between 0.8 and 0.99. Therefore, based on the advanced case, the experiment is carried out and the result is shown in Figure 2.

In the given scope of  $\beta$ , the variation curve of cost with the cooling schedule  $t = t / (1 + \beta t)$  is close to a parabolic curve, whose minimum could be reached at about 0.005. On the other side, in the given scope of  $\alpha$ , the variation curve of cost with the cooling schedule  $t = \alpha t$  is fluctuant, showing no obvious minimum. And the extremum is less than 700, which could be due to the stochastic error. So  $\alpha$  is set at about 0.97.

```

(1) Read the data file;
(2) NowPlan=Initial a plan;
(3) //The simulated annealing algorithm
(4) Initial the parameter;
(5) while ( $t > t\_limit$ ) do //outer circulation
(6)   while ( $n(T_k) > 0$ ) do //inner circulation
(7)     Generate a domain structure and a random number.
(8)     if (random < a1) then
(9)       NewPlan=Add a video content server;
(10)    else if (random < a2) then
(11)      NewPlan=Reduce a video content server;
(12)    else
(13)      NewPlan=Change a video content server;
(14)    end if
(15)    Calculate the min cost of the plan by ZKW algorithm.
(16)    Transmit the data of the graph into the ZKW algorithm to calculate
        the mincost of the current plan.
(17)    Calculate  $\Delta f = \text{NewPlanCost} - \text{NowPlanCost}$ .
(18)    if ( $\Delta f < 0$ ) then
(19)      NowPlan = NewPlan;
(20)    else
(21)      Calculate  $\exp(-\Delta f/T)$  and Generate a random number.
(22)      if ( $\exp(-\Delta f/T) > \text{random}$ ) then
(23)        NowPlan = NewPlan;
(24)      end if
(25)    end if
(26)     $n(T_k) - -$ ;
(27)  end while//inner circulation
(28)   $T = T/(1 + \beta * T)$ ; //Cool down the temperature.
(29)  if ( $T$  is low enough) then
(30)    set a large  $T$ .
(31)  end if
(32) end while//outer circulation
(33) //ZKW algorithm
(34) Initial the parameter;
(35) //method of bool
(36) Boolean the edges that still have flow but have not been visited.
(37) //method of augment
(38) Augment until it reaches the ending node, then recall and minus
        the flow of the visited nodes, and recurse and continue augmenting.
(39) //method of zkw
(40) while (Invoke the method of bool) do
(41)   while (Invoke the method of augment) do
(42)     for ( $i = 1$  to visited.length) do
(43)       visited[ $i$ ] = false;
(44)     end for
(45)   end while
(46) end while
(47) return mincost;

```

PSEUDOCODE 1

TABLE 1: Results of the proposed method under different cases.

Times	Primary case		Intermediate case		Advanced case	
	Iteration	Cost	Iteration	Cost	Iteration	Cost
(1)	176031	21986	106255	43344	11120	110088
(2)	177184	21986	104478	43322	10141	109900
(3)	175424	21986	109417	43322	10962	110051
(4)	173893	21986	107182	43342	10622	110022
(5)	174268	21986	108327	43322	10858	109769

TABLE 2: Comparison of Floyd, Dijkstra, and “ZKW” algorithms under the intermediate case.

Times	Floyd		Dijkstra		ZKW	
	Iteration	Cost	Iteration	Cost	Iteration	Cost
(1)	5	53918	167	45273	106806	43342
(2)	5	53242	150	45189	105254	43322
(3)	4	53884	140	45116	109857	43342
(4)	4	53696	147	45255	104425	43322
(5)	5	53994	143	44839	102656	43322

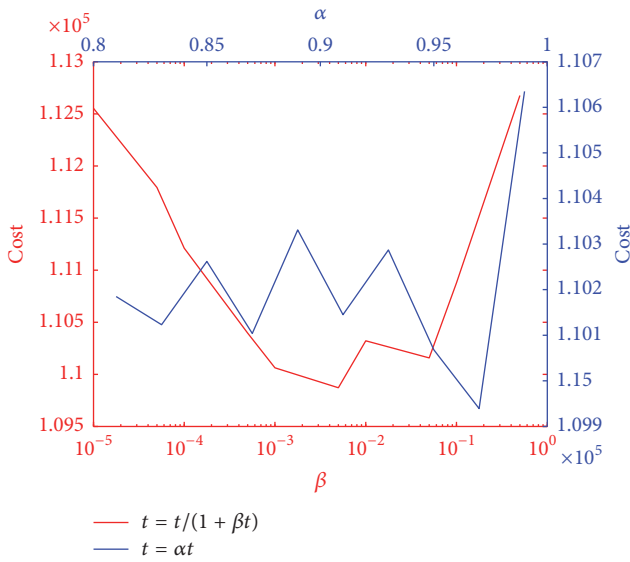


FIGURE 2: Cost under different  $\beta$  and  $\alpha$ .

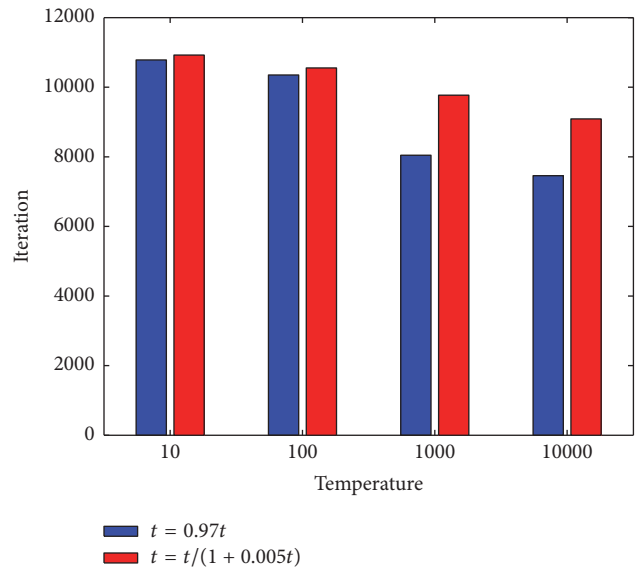


FIGURE 3: Iteration under different initial temperature and cooling schedule.

4.5. Comparison of Different Temperature Parameters. In order to find out a better temperature parameter of the simulated annealing algorithm, including the initial temperature and the cooling schedule, two cooling schedules  $t = 0.97t$  and  $t = t/(1 + 0.005t)$  are selected to run the program with the initial temperature  $T_0 = 10, 100, 1000, 10000$ . Based on the advanced case, it will be run for five times to obtain a mean value when the parameter is changed. And the result is given in Figures 3 and 4.

With the initial temperature increasing, the iterations of two cooling schedules decrease (see Figure 3). This is because, in a higher temperature, simulated annealing will search for the solution in a large solution space, which may cost much time. Although higher initial temperature may lead to the possibility of finding the global optimum, the actual result (see Figure 4) indicates that the cost is larger than a low initial temperature, especially with the cooling schedule  $t = 0.97t$ . Therefore, high initial temperature is not applicable for the advanced case.

Different cooling schedules will also have an impact on the performance of the algorithm. Apparently, the iteration of the schedule  $t = 0.97t$  is always lower than the schedule  $t = t/(1 + 0.005t)$ . In addition, the cost of the schedule  $t =$

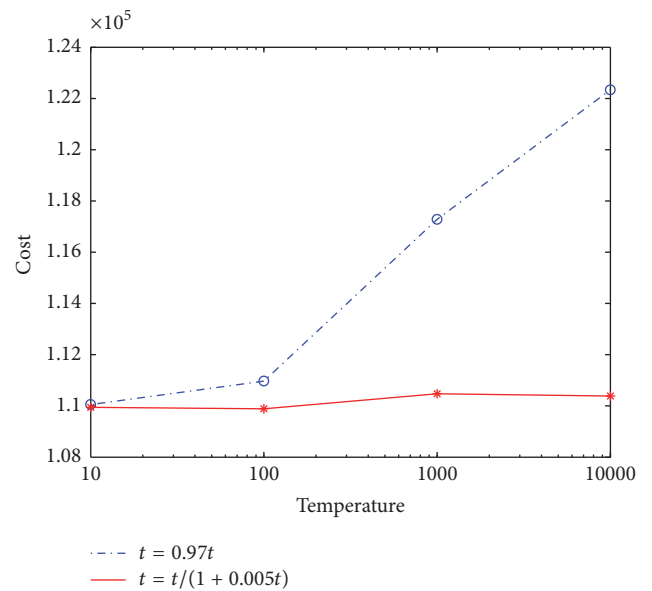


FIGURE 4: Cost under different initial temperature and cooling schedule.

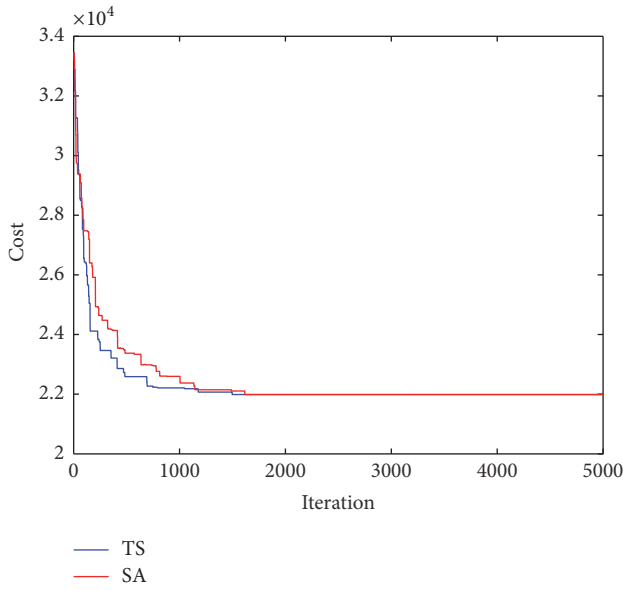


FIGURE 5: Comparison with tabu search under the primary case.

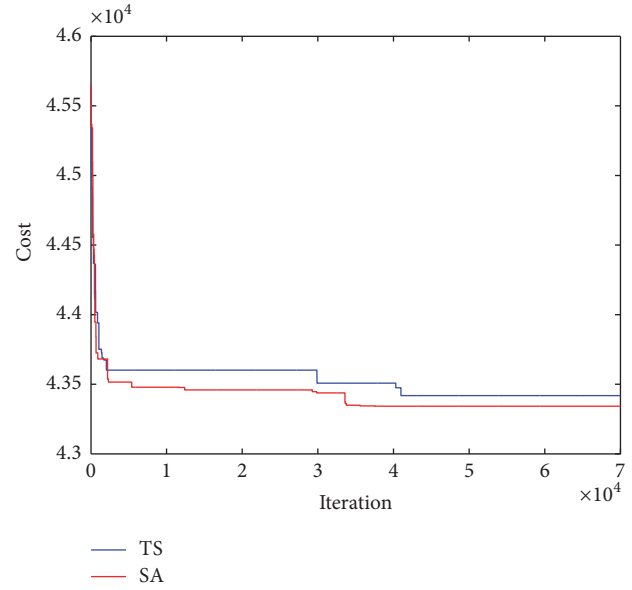


FIGURE 6: Comparison with tabu search under the intermediate case.

$0.97t$  is always larger than the schedule  $t = t/(1 + 0.005t)$ . What is more, the change of the initial temperature only has a little impact on the schedule  $t = t/(1 + 0.005t)$ , while the result of the cooling schedule  $t = 0.97t$  changes a lot. Thus, the cooling schedule  $t = t/(1 + 0.005t)$  is more stable and has strong robustness.

**4.6. Comparison with Tabu Search.** Since tabu search (TS) could balance the global and local searching, while simulated annealing mainly focuses on global searching, in this article, a comparison with tabu search is made to further discuss the proposed method. The searching processes are shown in Figure 5, 6, and 7.

Generally, simulated annealing has a better performance than tabu search under the intermediate case, while tabu search performs better under the advanced case. And both algorithms are able to find the global optimum when the scale of problem is small. This might be due to the different strategy for the algorithm to escape from the local optimum. Tabu search will only try to escape when it is trapped in the local optimum, while simulated annealing could escape at any time. Therefore, when the solution space is so large that the algorithms are unlikely to find the global optimum or have less chance to reach a more reasonable space to search, tabu search is more likely to reach a better solution than simulated annealing. On the contrary, when the solution space is not so large, like the intermediate case, the simulated annealing's strategy may be more suitable.

### 5. Conclusions

In this article, a method that combines simulated annealing algorithm with "ZKW" algorithm is proposed to solve the facility location problem. The method is applied to the cases given by HUAWEI Code Craft, which aims to find the appropriate network nodes to deploy video content server to

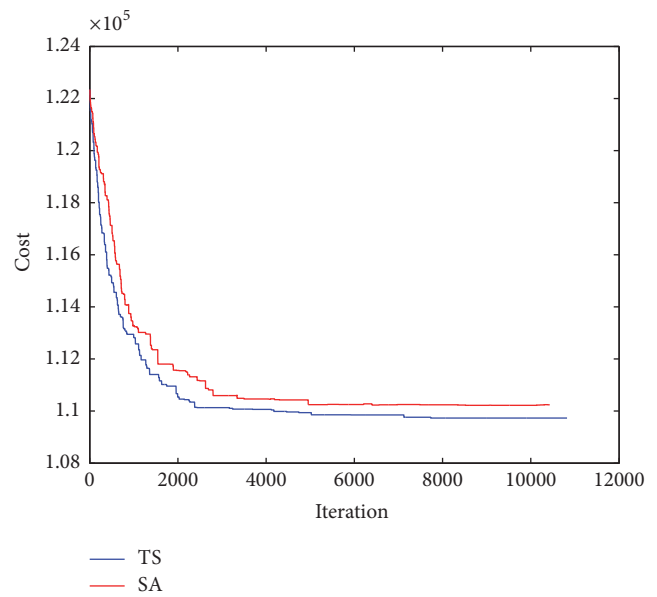


FIGURE 7: Comparison with tabu search under the advanced case.

satisfy the requirements of consumption nodes with the least cost in an undirected graph. By using the simulated annealing algorithm, we can easily find an optimum. Meanwhile, the "ZKW" algorithm makes it possible to find the shortest path and calculate the least cost with limited time.

The results of primary case, intermediate case, and advanced case illustrate the efficiency of the proposed method, which has not only a large iteration and less cost but also good robustness. Moreover, a comparison with traditional shortest path algorithms, Dijkstra and Floyd, shows that "ZKW" algorithm can greatly improve the speed of the program. In a limited amount of time, which is 90 s,

the running speed is quite important. The higher the running speed is, the more possibility for the simulated annealing to find the optimum there is. Besides, a comparison with tabu search shows that our method is more suitable for an intermediate scale of problem. In general, the proposed method that combines simulated annealing algorithm with “ZKW” algorithms performs pretty well in solving this video content server location problem.

### Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

### Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (Grants nos. 61573290 and 61503237).

### References

- [1] A. Weber and C. J. Friedrich, *Theory of the Location of Industries*, University of Chicago Press, Chicago, Ill, USA, 1962.
- [2] S. L. Hakimi, “Optimum locations of switching centers and the absolute centers and medians of a graph,” *Operations Research*, vol. 12, no. 3, pp. 450–459, 1964.
- [3] O. Kariv and S. L. Hakimi, “An algorithmic approach to network location problems. ii: The p-medians,” *SIAM Journal on Applied Mathematics*, vol. 37, no. 3, pp. 539–560, 1979.
- [4] S. H. Owen and M. S. Daskin, “Strategic facility location: a review,” *European Journal of Operational Research*, vol. 111, no. 3, pp. 423–447, 1998.
- [5] O. Berman, J. Kalcsics, and D. Krass, “On covering location problems on networks with edge demand,” *Computers and Operations Research*, vol. 74, pp. 214–227, 2016.
- [6] O. Berman, I. Hajizadeh, and D. Krass, “The maximum covering problem with travel time uncertainty,” *IIE Transactions (Institute of Industrial Engineers)*, vol. 45, no. 1, pp. 81–96, 2013.
- [7] I. Averbakh, O. Berman, D. Krass, J. Kalcsics, and S. Nickel, “Cooperative covering problems on networks,” *Networks. An International Journal*, vol. 63, no. 4, pp. 334–349, 2014.
- [8] S. M. Hedetniemi, E. J. Cockayne, and S. Hedetniemi, “Linear algorithms for finding the Jordan center and path center of a tree,” *Operations Research Society of America. Transportation Science Section. Transportation Science*, vol. 15, no. 2, pp. 98–114, 1981.
- [9] J. Brimberg, N. Mladenović, R. Todosijević, and D. Urošević, “General variable neighborhood search for the uncapacitated single allocation p-hub center problem,” *Optimization Letters*, vol. 11, no. 2, pp. 377–388, 2017.
- [10] O. Berman, N. Sanajian, and J. Wang, “Location choice and risk attitude of a decision maker,” *Omega (United Kingdom)*, vol. 66, pp. 170–181, 2017.
- [11] A. M. Geoffrion and G. W. Graves, “Multicommodity distribution system design by benders decomposition,” *Management Science*, vol. 20, no. 5, pp. 822–844, 1974.
- [12] H. Pirkul and V. Jayaraman, “A multi-commodity, multi-plant, capacitated facility location problem: formulation and efficient heuristic solution,” *Computers & Operations Research*, vol. 25, no. 10, pp. 869–878, 1998.
- [13] Z. Drezner and G. O. Wesolowsky, “Facility location when demand is time dependent,” *Naval Research Logistics*, vol. 38, no. 5, pp. 763–777, 1991.
- [14] W. Takahagi, Y. Sumitani, H. Takahashi, Y. Omae, and K. Sakai, “Method of determining future facility location with maintaining present accessibility,” *Industrial Engineering and Management Systems*, vol. 15, no. 3, pp. 197–205, 2016.
- [15] C. S. Revelle and G. Laporte, “The plant location problem: new models and research prospects,” *Operations Research*, vol. 44, no. 6, pp. 864–874, 1996.
- [16] C. Araz, H. Selim, and I. Ozkarahan, “A fuzzy multi-objective covering-based vehicle location model for emergency services,” *Computers and Operations Research*, vol. 34, no. 3, pp. 705–726, 2007.
- [17] A. Ghodrtnama, R. Tavakkoli-Moghaddam, and A. Azaron, “Robust and fuzzy goal programming optimization approaches for a novel multi-objective hub location-allocation problem: a supply chain overview,” *Applied Soft Computing Journal*, vol. 37, pp. 255–276, 2015.
- [18] T. Drezner, Z. Drezner, and P. J. Kalczynski, “The multiple markets competitive location problem,” *Kybernetes. The International Journal of Cybernetics, Systems and Management Sciences*, vol. 45, no. 6, pp. 854–865, 2016.
- [19] Y. Zhang, L. V. Snyder, T. K. Ralphs, and Z. Xue, “The competitive facility location problem under disruption risks,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 93, pp. 453–473, 2016.
- [20] Y. Du, X. Lu, X. Su, Y. Hu, and Y. Deng, “New failure mode and effects analysis: an evidential downscaling method,” *Quality and Reliability Engineering International*, vol. 32, no. 2, pp. 737–746, 2016.
- [21] R. Zhang, X. Ran, C. Wang, and Y. Deng, “Fuzzy evaluation of network vulnerability,” *Quality and Reliability Engineering International*, vol. 32, no. 5, pp. 1715–1730, 2016.
- [22] J. G. Kliniewicz, “A dual algorithm for the uncapacitated hub location problem,” *Location Science*, vol. 4, no. 3, pp. 173–184, 1996.
- [23] A. T. Ernst and M. Krishnamoorthy, “An exact solution approach based on shortest-paths for p-hub median problems,” *INFORMS Journal on Computing*, vol. 10, no. 2, pp. 149–162, 1998.
- [24] D. Valinsky, “Symposium on applications of operations research to urban services—a determination of the optimum location of fire-fighting units in New York City,” *Journal of the Operations Research Society of America*, vol. 3, no. 4, pp. 494–512, 1955.
- [25] R. C. Larson, “A hypercube queuing model for facility location and redistricting in urban emergency services,” *Computers & Operations Research*, vol. 1, no. 1, pp. 67–95, 1974.
- [26] Z. Drezner, V. Marianov, and G. O. Wesolowsky, “Maximizing the minimum cover probability by emergency facilities,” *Annals of Operations Research*, vol. 246, no. 1–2, pp. 349–362, 2016.
- [27] A. Ahmadi-Javid, P. Seyedi, and S. S. Syam, “A survey of health-care facility location,” *Computers and Operations Research*, vol. 79, pp. 223–263, 2017.
- [28] D. R. Dott, S. C. Wirasinghe, and A. Chakma, “Optimal hub location in pipeline networks,” in *Proceedings of the 1996 1st International Pipeline Conference, IPC. Part 1 (of 2)*, pp. 79–83, June 1996.
- [29] F. ÇEbi and İ. Otay, “Multi-Criteria and Multi-Stage Facility Location Selection under Interval Type-2 Fuzzy Environment: A Case Study for a Cement Factory,” *International Journal of*



- Computational Intelligence Systems*, vol. 8, no. 2, pp. 330–344, 2015.
- [30] R. Tavakkoli-Moghaddam, S. Sadri, N. Pourmohammad-Zia, and M. Mohammadi, “A hybrid fuzzy approach for the closed-loop supply chain network design under uncertainty,” *Journal of Intelligent and Fuzzy Systems*, vol. 28, no. 6, pp. 2811–2826, 2015.
- [31] X. Zhang, A. Adamatzky, X.-S. Yang, H. Yang, S. Mahadevan, and Y. Deng, “A Physarum-inspired approach to supply chain network design,” *Science China. Information Sciences*, vol. 59, no. 5, Article ID 052203, 052203, 14 pages, 2016.
- [32] M. Eskandarpour, P. Dejax, and O. Péton, “A large neighborhood search heuristic for supply chain network design,” *Computers and Operations Research*, vol. 80, pp. 23–37, 2017.
- [33] J. Liu, F. Lian, and M. Mallick, “Distributed compressed sensing based joint detection and tracking for multistatic radar system,” *Information Sciences. An International Journal*, vol. 369, pp. 100–118, 2016.
- [34] S. Kirkpatrick, J. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *American Association for the Advancement of Science. Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [35] S. Kirkpatrick, “Optimization by simulated annealing: quantitative studies,” *Journal of Statistical Physics*, vol. 34, no. 5–6, pp. 975–986, 1984.
- [36] P. Carnevali, L. Coletti, and S. Patarnello, “Image processing by simulated annealing,” *Journal of Research and Development*, vol. 29, no. 6, pp. 569–579, 1985.
- [37] M. Zhao, X. Yin, and H. Yue, “Genetic simulated annealing-based kernel vector quantization algorithm,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 31, no. 5, Article ID 1758002, 2017.
- [38] I. H. Osman, “Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem,” *Annals of Operations Research*, vol. 41, no. 4, pp. 421–451, 1993.
- [39] V. F. Yu, A. A. N. P. Redi, Y. A. Hidayat, and O. J. Wibowo, “A simulated annealing heuristic for the hybrid vehicle routing problem,” *Applied Soft Computing Journal*, vol. 53, pp. 119–132, 2017.
- [40] T. Loukil, J. Teghem, and P. Fortemps, “A multi-objective production scheduling case study solved by simulated annealing,” *European Journal of Operational Research*, vol. 179, no. 3, pp. 709–722, 2007.
- [41] L. M. R. Rere, M. I. Fanany, and A. M. Arymurthy, “Simulated annealing algorithm for deep learning,” *Procedia Computer Science*, vol. 72, pp. 137–144, 2015.
- [42] ZKW, “From point to cover: The minimum cost flow, ZKW algorithm,” <https://artofproblemsolving.com/community/c1368h1020435>.
- [43] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.
- [44] R. W. Floyd, “Algorithm 97: shortest path,” *Communications of the ACM*, vol. 5, no. 6, p. 345, 1962.
- [45] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, “Equation of state calculations by fast computing machines,” *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.
- [46] H. W. Kuhn, “The Hungarian method for the assignment problem,” *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.
- [47] H. W. Kuhn, “Variants of the Hungarian method for assignment problems,” *Naval Research Logistics Quarterly*, vol. 3, pp. 253–258 (1957), 1956.
- [48] J. Munkres, “Algorithms for the assignment and transportation problems,” *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957.
- [49] HUAWEI, “Code craft,” <http://codecraft.huawei.com/>.
- [50] H. Zhang, D. Wei, Y. Hu, X. Lan, and Y. Deng, “Modeling the self-similarity in complex networks based on Coulomb’s law,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 35, pp. 97–104, 2016.
- [51] W. Jiang, B. Wei, J. Zhan, C. Xie, and D. Zhou, “A visibility graph power averaging aggregation operator: a methodology based on network analysis,” *Computers Industrial Engineering*, vol. 101, pp. 260–268, 2016.
- [52] W. Jiang, B. Wei, Y. Tang, and D. Zhou, “Ordered visibility graph average aggregation operator: an application in produced water management,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 27, no. 2, p. 023117, 2017.
- [53] Y. Hu, F. Du, and H. L. Zhang, “Investigation of unsteady aerodynamics effects in cycloidal rotor using RANS solver,” *Aeronautical Journal*, vol. 120, no. 1228, pp. 956–970, 2016.
- [54] S. Wang, Y. Du, and Y. Deng, “A new measure of identifying influential nodes: efficiency centrality,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 47, pp. 151–163, 2017.
- [55] P. Amorim, E. Curcio, B. Almada-Lobo, A. P. Barbosa-Póvoa, and I. E. Grossmann, “Supplier selection in the processed food industry under uncertainty,” *European Journal of Operational Research*, vol. 252, no. 3, pp. 801–814, 2016.
- [56] X. Zhang, Y. Deng, F. T. S. Chan, P. Xu, S. Mahadevan, and Y. Hu, “IFSJSP: a novel methodology for the Job-Shop Scheduling Problem based on intuitionistic fuzzy sets,” *International Journal of Production Research*, vol. 51, no. 17, pp. 5100–5119, 2013.
- [57] H. Zheng, Y. Deng, and Y. Hu, “Fuzzy evidential influence diagram and its evaluation algorithm,” *Knowledge-Based Systems*, vol. 131, pp. 28–45, 2017.
- [58] H. Akyar, “Fuzzy risk analysis for a production system based on the Nagel point of a triangle,” *Mathematical Problems in Engineering*, Article ID 3080679, Art. ID 3080679, 9 pages, 2016.
- [59] W. Jiang and J. Zhan, “A modified combination rule in generalized evidence theory,” *Applied Intelligence*, vol. 46, no. 3, pp. 630–640, 2017.
- [60] X. Zhou, X. Deng, Y. Deng, and S. Mahadevan, “Dependence assessment in human reliability analysis based on D numbers and AHP,” *Nuclear Engineering and Design*, vol. 313, pp. 243–252, 2017.
- [61] X. Zhou, Y. Shi, X. Deng, and Y. Deng, “D-DEMATEL: A new method to identify critical success factors in emergency management,” *Safety Science*, vol. 91, pp. 93–104, 2017.
- [62] H. Mo and Y. Deng, “A new aggregating operator for linguistic information based on D numbers,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 24, no. 6, pp. 831–846, 2016.
- [63] X. Zhang, Y. Deng, F. T. Chan, and S. Mahadevan, “A fuzzy extended analytic network process-based approach for global supplier selection,” *Applied Intelligence*, vol. 43, no. 4, pp. 760–772, 2015.
- [64] X. Zhang, Y. Deng, F. T. Chan, A. Adamatzky, and S. Mahadevan, “Supplier selection based on evidence theory and analytic network process,” *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 2015.
- [65] M. Lundy and A. Mees, “Convergence of an annealing algorithm,” *Mathematical Programming*, vol. 34, no. 1, pp. 111–124, 1986.



# Hindawi

Submit your manuscripts at  
<https://www.hindawi.com>

