

## Research Article

# A Graph Based Framework for Route Optimization in Sea-Trade Logistics

Dianhui Chu,<sup>1</sup> Chunshan Li,<sup>1</sup> Xiaofei Xu,<sup>1</sup> and Xiaofeng Zhang<sup>2</sup>

<sup>1</sup>Department of Computer Science and Technology, Harbin Institute of Technology at Weihai, Weihai 264209, China

<sup>2</sup>Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, China

Correspondence should be addressed to Chunshan Li; [chunshanli@qq.com](mailto:chunshanli@qq.com)

Received 2 June 2015; Accepted 13 July 2015

Academic Editor: Xiaoyu Song

Copyright © 2015 Dianhui Chu 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.

How to find the optimal transportation route in sea-trade is very important for the logistics industry. The traditional routing problem is solved by performing the combinatorial optimization over a specified transportation network. Facing the huge network extracted from the foreign trading industry as well as the complex constraints, it is impossible for the traditional optimization methods to find the solution in a short time, which motivates our work. In this paper, we first carefully study the property of foreign trade network, and then convert the transportation network into a hierarchical one and propose a novel framework based on graphical model to solve this large scale network optimization problem. The experimental results demonstrate that our approach is superior to the famous ant colony optimization algorithm (ACO) in terms of accuracy and the time spent.

## 1. Introduction

With the rapid economic globalization, logistics industry has become a critical component in the commercial link. In addition to providing transportation services, modern logistics industry has brought many additional values into our society, such as electronic tracking, warehousing, and resources distribution. It consists of service center, information processing center, and resource allocation center. The logistics network is required to build seamless connection such that the aforementioned three main components can operate efficiently [1–3].

However, one important issue, path programming problem, aroused is how to find the optimal transportation route such that its transportation cost is minimal amidst all possible routes on the logistics network. The traditional approaches seek the help of combinatorial optimization performed on the complete transportation network. In nature, the problem is NP-hard, and thus those methods can only get the approximate solutions. With the logistics network growing hugely, for example, in sea-trade industry, as well as the increasing diverse constraints posed by law or user requirements, it becomes harder to acquire efficient and effective solutions via

traditional methods. Then some heuristic algorithms or intelligent agent based algorithms are proposed by researchers, such as genetic algorithm, ant colony algorithm, and immune algorithm. Unfortunately, the above methods are also problematic. Firstly, the assumption, held by those methods, is in doubt that all edges' costs of the logistics network are known. In fact, it is unachievable to exactly compute the cost of each edge of such large scale logistics network. Secondly, with the large amount of constraints, it is every difficult to directly optimize the entire network. Thirdly, due to the lack of adaptive learning ability, little knowledge could be learnt from the historical data, and thus, the traditional models fail to gradually improve the model performance.

Motivated by these problems, we propose, in this paper, a graph based framework to acquire the optimal route in logistics network. The main contributions of the proposed framework are on (1) proposing a graph based learning framework to handle the route selection problem in sea-trade logistics; meanwhile traditional approaches are almost heuristic algorithms or intelligent agent based algorithms; (2) defining time and cost constraints to deal with users' requirement in reality; (3) proposing an incremental algorithm which can utilize additional data to update route optimization model.

In this framework, hence, three algorithms are designed for different scenario, which are briefly listed as follows:

- (1) A random walk model based transportation route algorithm (RWTR) is proposed. It, without prior knowledge, calculates the probability of the edge using historical data of routes, and the edge probability represents how likely a route exists between those two nodes.
- (2) A constraints oriented transportation route algorithm (CTR) is proposed, which extends the RWTR algorithm and considers diverse customer requirements such as time constraint and price constraint.
- (3) An incremental transportation route algorithm (ICTR) is proposed as the adaptive version of the CTR algorithm and it automatically adjusts the model parameters according to the new operating data.

The rest of the paper is organized as follows. Section 2 reviews some related works. The route optimization problem is formulated, especially in sea-trade industry, in Section 3. Section 4 introduces aforementioned three routing algorithms. Experiments and evaluation results are demonstrated in Section 5. Section 6 concludes the paper.

## 2. Related Works

Route optimization in logistics network has become a widely researched topic in its own right over the years. A lot of research works have proposed this problem. In its infancy, Holland [4] adopted the genetic algorithm (GA) to this problem and the GA is an adaptive heuristic based search algorithm premised on the evolution of natural selections and genetic variations. This spirit was then widely adopted to optimize the logistics routes in many research works in [5–7].

Of late years, the evolutionary algorithms (EA) [8, 9] tried to optimize the route using similar techniques like inheritance, mutation, selection, and crossover. Immune algorithm is a variation of genetic algorithms imitating the immune system to solve the multimodal function optimization problem [10, 11]. Ant colony optimization (ACO) is then proposed, as an intelligent agent based technique, for combinatorial optimization problems, which mimics the foraging behavior of ants driven by sensing pheromone produced by other ants that successfully found foods. In ACO, a number of intelligent agents, virtual ants, were first constructed, and the agent released decaying information along the path it walked together what it found. Other agents chose the path having stronger pheromone and, after a long-term, the optimal route was automatically achieved [12–14].

Random walk algorithm is another important task related to this research which is an effective way to traverse graph nodes. Following its assumption, a walker randomly chooses its next visited node (among direct neighbors of current node) with certain probabilistic preference for each neighbor node [15]. Given initial nodes, the algorithm can produce a ranked node path with several steps of the random walk, which follows a desired probability distribution (transition probability matrix). Tong et al. propose a useful measure of

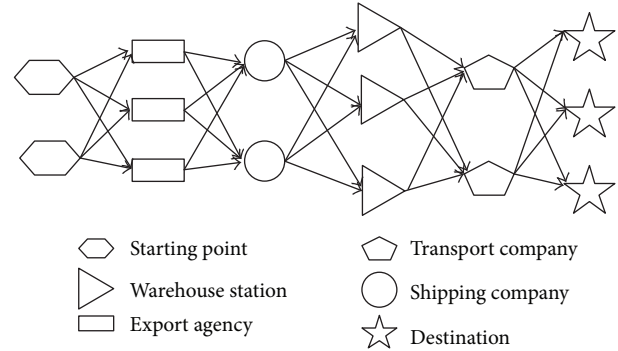


FIGURE 1: Logistics network of sea-trade.

node proximity based on graph topology [16]. Then, Fujiwara et al. propose a fast top-K search based on random walk algorithm, which uses BFS tree based pruning technique to skip unnecessary scanning of nodes for top-K results [17]. Recently, Yu and Lin [18] have developed an incremental algorithm that can update random walk algorithm on dynamical graphs.

Although those algorithms have been applied in a variety of domains such as graph coloring, routing selection, and the traveling salesman problem, they have several disadvantages for sea-trade, for example, slow convergence rate, high time complexity, less learning ability, and massive parameters to be learnt, which stems it from being applied in large scale applications.

## 3. Problem Formulation

Generally speaking, as seen in Figure 1, the logistics network of sea-trade includes six types of entities, for example, starting point (exporter), agency, shipping company, warehouse station, transportation company, and destination (importer). The business process of foreign is simplified as follows. Assume that an exporter at Weihai wants to sell his/her commodities to the USA through an agent. After finding the importer, the export agency employs a shipping company satisfying the requirements of exporter. Then, some warehouse stations are chosen to temporarily store the commodities. The transport company is in charge of transporting commodities from the warehouse station to the destinations.

According to the business process, the following characteristics can be extracted from the logistics network of sea-trade, which are given as follows:

- (1) Logistics network of sea-trade is a hierarchical one with each node belonging to a unique type which could be grouped by the same type layer.
- (2) Each layer has its unique position in the logistics chain.
- (3) A complete logistics route consists of edges sequentially connecting the adjacent layers.

With these characteristics, the logistics network can be modeled using Figure 1 and can be further abstracted as

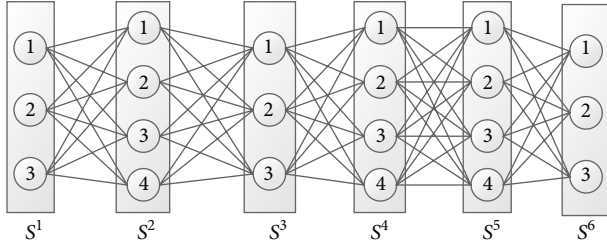


FIGURE 2: Abstracted network of sea-trade.

Figure 2. As shown in Figure 2, a logistics network is abstracted as a undirected graph  $G = \langle V, E \rangle$ , where  $V$  denotes the set of nodes and  $E$  denotes the set of edges. Node set  $V$  can be divided into  $k$  disjoint subsets:  $V = S^1 \cup S^2 \cup \dots \cup S^k$ , where  $S^i \cap S^j = \emptyset$ ,  $i, j \in \{1, \dots, k\}$ . Internal nodes in  $S^i$  are independent of each other. The incoming degree of all nodes in  $S^1$  is zero and the outdegree of all nodes in  $S^k$  is zero. A node is represented by  $v_i = (v_i^1, v_i^2, \dots, v_i^n)$ , where  $v_i^n \in S^i$ . Edge set is written as  $E = \{\langle v_i, v_j \rangle \mid v_i \in S^i, v_j \in S^{i+1}\}$ ,  $i \in \{1, \dots, k-1\}$ , and each edge,  $e = \langle v_i, v_j \rangle$ , has a weight  $w_{ij}$ , indicating the probability that  $e = \langle v_i, v_j \rangle$  exists in a logistics route. Therefore, the route optimization problem is, in essence, a ranking problem which can be decomposed into the following three subproblems:

- (1) Given a source node  $s \in S^1$ , a destination node  $d \in S^k$ , and the historical business data  $D$ ; how to output one optimal sequence  $\{r_2, \dots, r_n\}$  where  $r_i \in S^i$  ( $i = 2, \dots, k$ ).
- (2) Given a source node  $s \in S^1$ , a destination node  $d \in S^k$ , the historical business data  $D$ , and a set of constraints  $L = \{l_i(s_j) \mid s_j \in V - S^1\}$ ; how to output the optimal sequence  $\{r_2, \dots, r_n\}$  where  $r_i \in S^i$  ( $i = 2, \dots, k$ ), satisfying constraint set  $L$ .
- (3) Given a source node  $s \in S^1$ , a destination node  $d \in S^k$ , the historical business data  $D$ , a set of constraints  $L = \{l_i(s_j) \mid s_j \in V - S^1\}$ , and a new business data  $D_i$ ; how to output the optimal sequence  $\{r_2, \dots, r_n\}$ , where  $r_i \in S^i$  ( $i = 2, \dots, k$ ), which also satisfies constraint set  $L$ .

#### 4. The Proposed Algorithm

In this paper, the random walk model was adopted to optimize the logistics network. To make the analogy, the possibility to select next routing node can be viewed as the transition probability between the two nodes of the random walk process. Then, a transition probability matrix  $M$  of two adjacent node sets can be generated. Assume that a random path could be selected in the matrix  $M$ ; the model will gradually converge to its stable distribution [19], indicating the possibility that a route is chosen.

Let element  $v_j^i$  denote the probability that node  $j$  is chosen in the optimal route. Each pair  $\langle S^i, S^{i+1} \rangle$  in graph  $G$  generates a transition probability matrix  $M$ . Hence, there exist 5 transition probability matrices:  $M_1, M_2, M_3, M_4$ , and  $M_5$ .

Matrix  $M_i$  is the transition probability matrix on  $\langle S^i, S^{i+1} \rangle$ . The iterative equations to optimize the route are defined as follows:

$$\begin{aligned}
 v^1 &= (1-c)M_1v^2 + cp, \\
 v^2 &= \frac{1}{2}M_1^T v^1 + \frac{1}{2}M_2v^3, \\
 v^3 &= \frac{1}{2}M_2^T v^2 + \frac{1}{2}M_3v^4, \\
 v^4 &= \frac{1}{2}M_3^T v^3 + \frac{1}{2}M_4v^5, \\
 v^5 &= \frac{1}{2}M_4^T v^4 + \frac{1}{2}M_5v^6, \\
 v^6 &= (1-c)M_5^T v^6 + cq,
 \end{aligned} \tag{1}$$

where vectors  $p$  and  $q$  are the initial value and  $c$  is a constant which is empirically set to 0.5.

In (1), the transition probability matrices,  $M_1, M_2, M_3, M_4$ , and  $M_5$ , are unknown and should be calculated first. They can be calculated by the following procedure using the historical data. The weight of edge is calculated as  $w_{ij} = f_\psi(v_i, v_j)$ , where  $f_\psi$  is used to compute the transition probability between  $v_i$  and  $v_j$ .

Let  $F(v_i, v_j, \psi) = -\sum_{k=1}^n \psi_k \text{sim}(v_i^k, v_j^k)$  be the similarity between  $v_i$  and  $v_j$ , with  $\text{sim}(v_i^k, v_j^k)$  quantifying similarity between  $v_i$  and  $v_j$  on the  $k$ th-dimension. Then the transition probability can be defined as

$$f_\psi(v_i, v_j) = \frac{1}{1 + \exp(-\sum_{k=1}^n \psi_k * \text{sim}(v_i^k, v_j^k))}. \tag{2}$$

The parameter  $\psi$  can be estimated using the maximum likelihood. The likelihood function is  $l(\psi) = \log(\prod_m f_\psi(v_i, v_j))$ , where  $m$  is number of edges. The maximization step can be derived as

$$\begin{aligned}
 \frac{\partial F(v_i, v_j, \psi)}{\partial \psi_k} &= \text{sim}(v_i^k, v_j^k), \\
 \frac{\partial L}{\partial \psi_k} &= (-1) \sum_m \frac{\exp(-F(v_i, v_j, \psi))}{1 + \exp(-F(v_i, v_j, \psi))} \frac{\partial F(v_i, v_j, \psi)}{\partial \psi_k} \\
 &= (-1) \sum_m \frac{\exp(-F(v_i, v_j, \psi))}{1 + \exp(-F(v_i, v_j, \psi))} \text{sim}(v_i^k, v_j^k),
 \end{aligned} \tag{3}$$

$$\psi_t = \psi_{t-1} + \eta \frac{\partial L}{\partial \psi},$$

where  $\eta$  is iterative parameter, and the iteration will stop when the difference  $|\psi_t - \psi_{t-1}|$  is smaller than a predefined value  $\epsilon$ .

With (1), (2), and (3), all transition probability matrix could be acquired. Now, we will introduce the proposed algorithms, RWTR, CTP, and ICTR, respectively. When there are no special requirements (or constraints), the RWTR algorithm can be adopted to acquire the optimal route.

As this algorithm directly adopts the random walk model to optimize the route, we just give the details in Algorithm 1.

*Algorithm 1* (RWTR algorithm).

*Input.*

Properties of exporter commodity  $P$   
Business dataset  $D$ .

*Output.* A route in graph  $G$  and transition probability matrix

- (1) generate transition probability matrix  $M_1, M_2, M_3, M_4, M_5$
- (2) according to  $P$ , generate the vector  $p$  and  $q$ ;
- (3) initialize vector  $v^1, v^2, v^3, v^4, v^5, v^6$
- (4) **while** *isNotConvergence* ( $v^1, v^2, v^3, v^4, v^5, v^6$ ) **do**
- (5) compute (1)
- (6) **end while**
- (7) obtain index of  $k$ th-dimension in  $v^i, i \in \{2, 3, 4, 5\}$ , where the possibility of  $i$ th-dimension is maximum.
- (8) obtain route is make up of source and destination and indexes in  $v^2, v^3, v^4, v^5$ .

When constraints, such as time spent in the transportation and price to be charged, are considered, the RWTR was extended to the CTR algorithm. In this algorithm, the global constraint can be segmented into several fragments which is based on the statistical estimation on its historical data. The number of those fragments is set to the number of layers in the network with each constraint fragment corresponding to a layer. For example, in the statistics of historical information, if the average time the network takes to transport the same goods is ten days and five days are spent on warehouse station, then the percentage of the layer of warehouse station is initialized as 50% when we consider the time constraint. All layers are initialized according to their historical statistics. Similarly, the remaining constraints could be processed. The model learning part is slightly revised and details of the CTR are given in Algorithm 2.

*Algorithm 2* (CTR algorithm).

*Input.*

Properties of exporter commodity  $P$   
transition probability matrix  $M_1, M_2, M_3, M_4, M_5$   
Constraints of  $L = \{l_i(s_j) \mid s_j \in V - S^1 - S^k\}$   
Business dataset  $D$ .

*Output.* A route in graph  $G$ :

- (1) according to  $P$ , generate the vector  $p$  and  $q$ ;
- (2) initialize vector  $v^1, v^2, v^3, v^4, v^5, v^6$
- (3) **while** *isNotConvergence* ( $v^1, v^2, v^3, v^4, v^5, v^6$ ) **do**
- (4) compute (1)

(5) **end while**

- (6) obtain index of  $k$ th-dimension in  $v^i, i \in \{2, 3, 4, 5\}$ , where the possibility of  $i$ th-dimension is descending order and the  $k$ th-node satisfies the constraint  $L_i$
- (7) obtain route is make up of source and destination and indexes in  $v^2, v^3, v^4, v^5$ .

To further consider the effect of incoming data, the ICTR algorithm is proposed, shown in Algorithm 3. Once the incremental data set  $D_t$  is received, the transition probability matrices  $M_1, M_2, M_3, M_4$ , and  $M_5$  are updated by (4) immediately, which is given as

$$M'_i = \frac{|D|}{|D| + |D_t|} M_i + \frac{|D_t|}{|D| + |D_t|} M_{it}. \quad (4)$$

*Algorithm 3* (ICTR algorithm).

*Input.*

Properties of exporter commodity  $P$   
transition probability matrix  $M_1, M_2, M_3, M_4, M_5$   
Constraints of  $L = \{l_i(s_j) \mid s_j \in V - S^1 - S^k\}$   
Business data set  $D$  and New business data  $D_t$ .

*Output.* A route in graph  $G$

- (1) generate transition probability matrix  $M'_1, M'_2, M'_3, M'_4, M'_5$  by (4)
- (2) according to  $P$ , generate the vector  $p$  and  $q$ ;
- (3) initialize vector  $v^1, v^2, v^3, v^4, v^5, v^6$
- (4) **while** *isNotConvergence* ( $v^1, v^2, v^3, v^4, v^5, v^6$ ) **do**
- (5) compute (1)
- (6) **end while**
- (7) obtain index of  $k$ th-dimension in  $v^i, i \in \{2, 3, 4, 5\}$ , where the possibility of  $i$ th-dimension is descending order and the  $k$ th-node satisfies the constraint  $L_i$
- (8) obtain route is make up of source and destination and indexes in  $v^2, v^3, v^4, v^5$ .

## 5. Experiments Analysis

In this section, a set of synthetic data sets are constructed for the evaluation of the proposed RWTR, CTR, and ICTR algorithms. The experimental results showed that the optimized route acquired by our methods is superior to those acquired by the baseline algorithms with respect to total price and total time spent.

*5.1. Experiment Design.* As there is no benchmark data set, a set of synthetic data sets will be generated as follows. The network, shown in Figure 2, is simulated. The nodes in the network are generated as (1) each node is generated with 5 attributes, shown in Table 1, and these attributes are considered in this model as they are commonly accepted as the key

TABLE 1: Node attributes.

Attribute	Description	Scope
Location	Location of nodes	1-5
Destination	Destination of goods (in station layer, it means the location of station)	1-5
Weight	The weight of the goods or the maximum ability of dealing with goods	1-5
Time constraint	The minimum time of dealing with issues (in starting point, it means the time constraint needed by exporter)	1-5
Price	The cost for going through a node	1-5

TABLE 2: Calculation formula for attribute distances.

Attribute	Formula
Location	$\begin{cases} 0 & \text{where } v_{ik} = v_{jk} \\ 0.2 & \text{where } v_{ik} \neq v_{jk} \end{cases}$
Destination	$\begin{cases} 1 & \text{where } v_{ik} \neq v_{jk} \\ 0 & \text{where } v_{ik} = v_{jk} \end{cases}$
Weight	$\begin{cases}  v_{ik} - v_{jk}  & \text{other} \\ \infty & \text{where } v_{ik} > v_{jk} \end{cases}$
Time constraint	$\begin{cases}  v_{ik} - \theta_k  & \text{other} \\ \infty & \text{where } v_{ik} > \theta_n \end{cases}$ <p style="text-align: right;">where <math>\theta_i</math> is the time constraint of layer <math>i</math></p>
Price	$\frac{1}{v_{jk}}$

factors affecting the selection of route and (2) the attributes belonging to the node in the layer of starting point need to reflect some constraints posed by the customers, which make them different from nodes of other layers. For example, the time constraint attribute of the nodes, in starting point layer, indicates the delivery time of the goods, whereas it only means, in the remaining layers, the processing time a node takes. In starting point layer, weight attribute is thought as the weight of goods, but it means, in other layers, the maximum weight could be processed by the node. The price attributes are usually set to zero as exporters focus more on other constraints, which can be retrieved in nodes of other types. In the rest of this paper, node, not in the starting point layer or destination layer, will be called internode and its layer is called interlayer.

In the generated network, there are 5 locations, 5 destinations, and 10 internodes. The attributes of these nodes are illustrated in Table 1. To eliminate the ambiguousness of the transition probability matrix, each node in the start point layer is duplicated 4 times, with each node reflecting one constraint. Then, the time constraint attribute will be classified into 2 classes, namely, short-term and long-term, normalized by 10. The weight attribute will also be split into 2 classes, namely, heavy goods and light goods, normalized by 50. Therefore, the combination of requirements on the location is total of 4 different categories, that is, short-term, heavy goods, long-term, heavy goods, short-term, light goods, and short-term, light goods. To match these 4 categories, the duplicated nodes are generated. In the end, the synthetic network consists of 65 nodes (20 starting nodes, 5 destination nodes, and 40 internodes).

Two data sets are generated based on previous network, the first one does not consider the time constraints, and 70 routes are selected. Among them, 50 routes are randomly selected out for the training of parameter  $\psi$ , and the remaining 20 routes will be used to test the performance of the RWTR. The second data set is generated to consider the time constraint, in which 100 routes are manually created. Each node in starting point layer has at least one route. A random sample of 50 routes is chosen to train parameter  $\psi$ , a random sample of 20 routes is chosen to test the performance of the CTR, and the remaining 30 routes are to test the performance of the ICTR.

To evaluate the model performance, the ACO is chosen as the baseline algorithm for the comparison. The ACO is a probabilistic approach which can find the optimal paths through its self-learning process. We first describe the ACO in Algorithm 4. In the ACO, the quality of the path of the ant is inversely proportional to the objective function value of ACO. The objective function of ACO can be defined as  $f(\text{route}) = \sum_{i=1} \sum_{j=1} F(v_i, v_j, \psi)$ , where  $F(v_i, v_j, \psi)$  is defined in (2). Parameters of the ACO are set as follows. The number of ants is 60, the maximum number of cycles is set to 300,  $\alpha = 0.5$ ,  $\beta = 1$ , and  $\rho = 0.7$ . As Euclidean distance is not applicable, Table 2 shows how we calculate the distance in predefined attributes, in which  $v_{ik}$  is the  $k$ th component of  $v_i$ , and  $v_i$  is the predecessor node of  $v_j$  in the network. For attribute of location, if the two nodes belong to the same city, the distance between two nodes is 0; otherwise the distance is 0.2. For attribute of destination, if the export destination of two nodes is the same in sea-trade, the distance between two nodes is 0;

TABLE 3: Evaluation for parameters of ACO.

Type	Correct route number
Location first	11
Destination first	11
Weight first	12
Time constraint first	14
Price first	12
Attributes equally	11

otherwise the distance is 1. For attribute of weight, if the maximum loading weight of predecessor node  $v_i$  is larger than node  $v_j$  in sea-trade, then the merchandise cannot be transported from  $v_i$  to  $v_j$ ; the distance between two nodes is  $\infty$ ; otherwise the distance is  $|v_{ik} - v_{jk}|$ . For attribute of time constraint, if the time cost in successor node  $v_j$  is larger than time limitation of  $\theta_k$ , then the path between  $v_i$  and  $v_j$  is unavailable, and then the distance between two nodes is  $\infty$ ; otherwise the distance is  $|v_{ik} - \theta_k|$ . For attribute of price, the distance between  $v_i$  and  $v_j$  is inversely proportional to cost in node  $v_i$ .

*Algorithm 4* (ACO algorithm in logistics).

*Input.*

Properties of exporter commodity  $P$

Constraints of  $L = \{l_i(s_j) \mid s_j \in V - S^1 - S^k\}$

Business dataset  $D$ .

*Output.* A route in graph  $G$  and transition probability matrix

- (1) **while** count (circle) < maxCircleCount **do**
- (2) all ants complete parades
- (3) update pheromone on the relative path, based on the quality of ant parade.
- (4) **end while**
- (5) obtain route is make up of edges on which the number of ants is more than other edges.

*5.2. Experimental Evaluation for Parameters of ACO.* To achieve the best performance of the ACO, parameters are first carefully tuned with a focus on parameters  $\psi$  as it directly affects the objective function of ACO. In this evaluation, ACO is performed on the second data set and  $\psi$  can be estimated on two scenarios: (1) attributes with priority:  $\psi_i = 0.6$  and  $\psi_{\text{other}} = 0.1$ , where  $i = 1, 2, \dots, 5$ , which means that the  $i$ th attribute is the most important attribute to the ants and (2) equally weighted attributes:  $\psi = [0.2, 0.2, 0.2, 0.2, 0.2]$ , which means all attributes are of equal importance to the ants. The results can be seen in Table 3. The first scenario achieves better performance than that of the second scenario, as the ants are able to achieve a feasible route while paying less attention to constraints like “weight.” Hence,  $\psi = [0.1, 0.1, 0.2, 0.4, 0.2]$  is used in the remaining experiments.

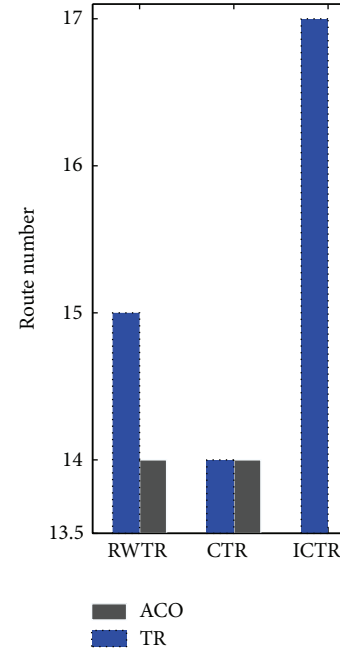


FIGURE 3: Accuracy of the RWTR, CTR, and ICTR.

### 5.3. Experimental Results of the RWTR, CTR, and ICTR

*5.3.1. Experimental Results of the RWTR.* In this evaluation, the RWTR algorithm is performed on the first data set and the parameter  $\psi$  is estimated using (3), and we have  $\psi = [0.005, 0.5608, 0.0732, 0.0683, 0.2477]$ . From the weight of each attribute, it can conclude that destination is the most important factor, the price attribute is the second important factor, the location is the least important factor, and the remaining two attributes are almost equal weighted. This roughly matches customers’ intuitive way to select the route. After acquiring  $\psi$ , all transition probability matrix can be calculated. The results of the RWTR can then be achieved. The ACO algorithm is directly performed on the test set.

From the comparison results in Figure 3, it was found that the RWTR algorithm can get 15 correct optimal routes, judged manually, in test data set, while the ACO algorithm can find 14 correct optimal routes. The accuracy of the RWTR is higher than the ACO by 71%. More importantly, the RWTR algorithm is 10 times faster than the ACO algorithm, and the speedup result is shown in Table 4. The reason lies in the fast convergence rate of the random work model.

*5.3.2. Experimental Result of the CTR.* In this subsection, the CTR algorithm is evaluated on the second data set. Similarly, the parameter  $\psi$  is calculated first, and the importance priority is in descendent order as “destination, weight, time constraint, price, and location,” in which the time constraint becomes more important than the price. After the learning of  $\hat{A}$ , the transition matrix is computed and the CTR results can be achieved. The ACO is performed on the test set, and the results are shown in Figure 3. This time, the CTR algorithm can obtain 14 correct optimal routes in test data set, while the

TABLE 4: Running time.

	RWTR	CTR	ICTR	ACO without constraint	ACO with constraint
Time	0.1032	0.1094	0.1087	1.2026	1.7832

ACO algorithm also finds 14 correct optimal routes. However, the CTR algorithm is 16 times faster than the ACO algorithm, which is shown in Table 4.

5.3.3. *Experimental Result of the ICTR.* Similarly, the ICTR algorithm is evaluated on the second data set by considering both the time constraints and the incoming data. The calculated parameter  $\psi$  indicates the importance priority of the attributes is the same as that of the CTR.

From the comparison results in Figure 3, we can see that ICTR can acquire 17 correct optimal routes in test data set, while the ACO algorithm finds only 14 correct optimal routes. The ICTR algorithm is also 16 times faster than the ACO algorithm. The improvement of the model accuracy lies in the knowledge acquired from the incoming data, which indicates that the ICTR is the best choice among all three approaches proposed.

## 6. Conclusions

In this paper, we have proposed a novel framework to find the optimal sea-trade route in the logistics network. After a careful study on the characteristics of the logistics network, the route optimization problem is decomposed into several subproblems, which is modeled as a hierarchical graph. With this graph, the random walk model was adopted, and on top of this, two extensions are proposed to consider the time constraints and the effect of the incoming data, respectively. By comparing with the ant colony learning algorithm, our algorithms can achieve better routes but consume much fewer time than that of the ACO. In the future, we will investigate how to further improve the model performance in a distributed manner.

## Conflict of Interests

The authors declare that they do not have any conflict of interests in their submitted paper.

## Acknowledgments

This work is supported in part by National Key Technology R&D Program (no. 2015BAF23B03), Research Award Foundation for Outstanding Young Scientists in Shandong Province, Grant no. 2014BSA10016, the Scientific Research Foundation of Harbin Institute of Technology at Weihai (Grant no. 201412), and the Fundamental Research Funds for the Central Universities (Grant no. HIT.NSRIF.2016104).

## References

- [1] H. K. H. Chow, K. L. Choy, W. B. Lee, and F. T. S. Chan, "Design of a knowledge-based logistics strategy system," *Expert Systems with Applications*, vol. 29, no. 2, pp. 272–290, 2005.
- [2] M. Goetschalckx, C. J. Vidal, and K. Dogan, "Modeling and design of global logistics systems: a review of integrated strategic and tactical models and design algorithms," *European Journal of Operational Research*, vol. 143, no. 1, pp. 1–18, 2002.
- [3] V. Jayaraman and H. Pirkul, "Planning and coordination of production and distribution facilities for multiple commodities," *European Journal of Operational Research*, vol. 133, no. 2, pp. 394–408, 2001.
- [4] J. H. Holland, "Genetic algorithms," *Scientific American*, vol. 267, no. 1, pp. 66–72, 1992.
- [5] B. Elbenani, J. A. Ferland, and J. Bellemare, "Genetic algorithm and large neighbourhood search to solve the cell formation problem," *Expert Systems with Applications*, vol. 39, no. 3, pp. 2408–2414, 2012.
- [6] Y. Rahmat-Samii and E. Michielssen, *Electromagnetic Optimization by Genetic Algorithms*, 1999.
- [7] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: formulation, discussion and generalization," in *Proceedings of the 5th International Conference on Genetic Algorithms*, pp. 416–423, Champaign, Ill, USA, June 1993.
- [8] P. J. Angeline, G. M. Saunders, and J. B. Pollack, "Evolutionary algorithm that constructs recurrent neural networks," *IEEE Transactions on Neural Networks*, vol. 5, no. 1, pp. 54–64, 1994.
- [9] M. J. Gacto, R. Alcalá, and F. Herrera, "A multi-objective evolutionary algorithm for an effective tuning of fuzzy logic controllers in heating, ventilating and air conditioning systems," *Applied Intelligence*, vol. 36, no. 2, pp. 330–347, 2012.
- [10] J.-S. Chun, M.-K. Kim, H.-K. Jung, and S.-K. Hong, "Shape optimization of electromagnetic devices using immune algorithm," *IEEE Transactions on Magnetics*, vol. 33, no. 2, pp. 1876–1879, 1997.
- [11] M. Gong, L. Jiao, H. Du, and L. Bo, "Multiobjective immune algorithm with nondominated neighbor-based selection," *Evolutionary Computation*, vol. 16, no. 2, pp. 225–255, 2008.
- [12] M. Dorigo, M. Birattari, and T. Stützle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.
- [13] M. Dorigo, M. A. Montes de Oca, S. Oliveira, and T. Stützle, *Ant Colony Optimization*, Wiley Encyclopedia of Operations Research and Management Science, 2007.
- [14] M. Dorigo and G. Di Caro, "Ant colony optimization: a new meta-heuristic," in *Proceedings of the Congress on Evolutionary Computation*, pp. 1470–1477, July 1999.
- [15] M. Zhong and K. Shen, "Random walk based node sampling in self-organizing networks," *ACM SIGOPS Operating Systems Review*, vol. 40, no. 3, pp. 49–55, 2006.
- [16] H. Tong, C. Faloutsos, and J.-Y. Pan, "Fast random walk with restart and its applications," in *Proceedings of the IEEE 6th International Conference on Data Mining (ICDM '06)*, pp. 613–622, December 2006.
- [17] Y. Fujiwara, M. Nakatsuji, M. Onizuka, and M. Kitsuregawa, "Fast and exact top-k search for random walk with restart," *Proceedings of the VLDB Endowment*, vol. 5, no. 5, pp. 442–453, 2012.

- [18] W. Yu and X. Lin, “IRWR: incremental random walk with restart,” in *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '13)*, pp. 1017–1020, 2013.
- [19] L. Lovász, “Random walks on graphs: a survey,” in *Combinatorics, Paul Erdos is Eighty*, vol. 2, pp. 1–46, 1993.





# Hindawi

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

