Enhancing market service and enterprise operations through a large-scale GIS-based distribution system

Wei Gu\textsuperscript{a}, Krista Foster\textsuperscript{b}, Jennifer Shang\textsuperscript{b,∗}

\textsuperscript{a} Donlinks School of Economics and Management, University of Science and Technology Beijing, Beijing 100083, China
\textsuperscript{b} Joseph M. Katz Graduate School of Business, University of Pittsburgh, Pittsburgh, PA 15260, United States

\section*{A R T I C L E   I N F O}

Keywords:
Decision support systems
Geographical information system
Marketing service
Large-scale logistics distribution

\section*{A B S T R A C T}

We develop a large-scale service information system integrating market analysis, customer relationship management (CRM), and distribution service optimization. The system is based on geographical information systems (GIS), and the goals include reducing the distribution cost, increasing the efficiency, satisfying customer demand, and improving service quality. We discuss in detail the design of the model and the implementation of the system. The main contributions of this work are: (1) proposing a new workload evaluation method based on statistical analysis of the large data set. The workload measure is based on GIS and enterprise databases, which address the workload imbalance issue in distribution; (2) implementing an optimal distribution model to serve nearly a hundred thousand retailers. The model contains two stages and uses a Cluster-First-Route-Second approach. In the clustering stage, we improve the K-means method; while at the routing stage, we design a hybrid heuristic algorithm for GIS data by employing the genetic algorithm and simulated annealing techniques; and (3) integrating market analysis, CRM, and distribution service optimization to improve market service and enterprise operations.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

The rapid development of informatics and the continued popularity of the Internet have driven the growth of e-commerce. As e-commerce develops, the number of customers or clients increases. High-frequency distribution and quick-response logistics are necessary for effective product delivery and customer service. Timely and accurate order fulfillment is thus vital for e-commerce success, but it poses a significant challenge to businesses. In developing countries, especially in China, the most populated country on earth, the customer base is huge but its logistics performance is less than perfect. To improve logistics efficiency and effectiveness, we develop a large-scale decision support system to optimize marketing service and enterprise operations. The information system integrates market analysis, customer relationship management (CRM), and distribution resource planning using data from geographical information systems (GIS). We strive to reduce costs, meet customer expectations of quality, excel in responsiveness, and improve operations efficiency and flexibility.

This research was motivated by the needs of a fast-moving-consumer-goods (FMCG) supply chain (SC) that ships its products to more than 90,000 retailers, which serve 24 million consumers dispersed throughout Inner Mongolia in Northwestern China. Annually, more than 1,150,000 customer orders and 13 million cases of products are shipped with $2.82 billion revenue. FMCG are characterized by short shelf life (sold quickly) and low cost, e.g. soft drinks and over-the-counter (OTC) non-prescription drugs. The distribution center faces many challenges. Currently, the company uses human judgment based on experience and instinct for distribution management and work allocation. Due to the complexity, the resources and distribution schedules remain nearly fixed once the company decides on them. However, customers’ demands and locations change from time to time, the types of delivery vehicles and goods may be mismatched, and deliverymen’s and drivers’ experiences vary. All these lead to waste, workload imbalance, and high distribution costs. The company is in need of a more sophisticated and practicable information system to optimize the logistics distribution process.

Fig. 1 shows the distribution of customers in Inner Mongolia and Baotou City. The retailers (the green diamonds) are dispersed in 14 cities in Inner Mongolia. Our study focuses on Baotou City. It is located in the east of Inner Mongolia. The city spans 28,000 km\textsuperscript{2}, and there are nine districts. The resident population in Baotou is 2.55 million, including 1.8 million urban residents. There is one regional distribution center in the downtown area with five customer transfer stations (the red circles). In total, 9494 retailers...
demand 14,000 cases of products annually. The methods developed for Baotou City in this research are generalizable and are used for other cities as well.

We develop the large-scale distribution support system in order to: (1) reduce distribution costs; (2) resolve the workload imbalance problem among different clusters; (3) improve customer service through marketing, delivery, and quick-response; and (4) build a GIS-based large-scale logistics distribution system, which integrates market analysis, CRM, and distribution requirement planning.

We make use of the powerful geospatial data analysis capabilities of GIS for logistics planning. It mainly processes and analyzes space. Our model applies dynamic and real-time GIS data to optimize distribution system. The solution is practical and can be viewed on the electronic map. However, there are some issues which must be addressed in the application of the GIS technology. First, collecting and updating all geographical data attributes for all customers is complicated. Second, decisions regarding how to integrate the GIS spatial analysis with the distribution model must be made. Third, the solution must be evaluated. In Sections 4 and 5, we detail the process of customer clustering and vehicle routing based on GIS data.

The distribution process under consideration in this paper has the following characteristics:

1. One distribution center serves multiple customers.
2. Thousands of customers are widely dispersed geographically.
3. Various types of vehicles with different capacities are available.
4. Customer demands are known.
5. Order quantities are small, yet customers reorder frequently.
6. The objective is to minimize the total distribution costs.
7. The distribution center is remotely located from the delivery area.

The solution is derived following the two steps below:

(i) Divide the large-scale complex distribution network into several small-scale distribution areas subject to certain constraints to obtain an initial feasible solution of the two-stage method. Through clustering, we transform the multi-vehicle optimization problem into several traveling salesman problems (TSPs).

(ii) Design a heuristic algorithm to solve the TSPs one by one for each service area, resulting in quasi-optimal solution to the entire distribution area.

The solution scheme is summarized in Fig. 2, including the following steps:

1. Analyze the current distribution process.
2. Determine the measurement of the workload, which helps design a true balanced workload among regions.
3. Calculate the shortest path between any two retail outlets using GIS.
4. Divide the distribution area into sub-regions by implementing a GIS network, and ensure workload balance.
(5) Plan the routes. As the number of outlets, number of orders, road congestion and other parameters are not deterministic, the TSP is solved based on the GIS network and the customer order database. The delivery route is dynamic and affected by traffic and orders.

The contributions of this research are five-fold:

1. We propose a more accurate approach to measure workloads. Specifically, we define workload as working time, including travel time and service time.
2. We propose an efficient framework for large-scale customer clustering and routing, based on an in-depth understanding of the firm, practical needs, and system constraints.
3. We develop a GIS-based partition method with the goal of balancing the workload assigned to each sub-area. This is achieved by adapting the two-stage K-means clustering method.
4. We develop a real-time dynamic GIS-based distribution schedule by taking advantage of the strengths of simulated annealing and the genetic algorithm.
5. We integrate the information system with market analysis, CRM, and distribution optimization.

The remainder of the paper is organized as follows. We begin with a literature review in Section 2, and discuss the system design in Section 3. Section 4 uses the K-means method to partition the distribution area by balancing the workload. A hybrid optimization algorithm is proposed in Section 5 to solve the GIS-based routing problem. Section 6 implements the decision support system for FMCG distribution. Finally, we present our conclusions in Section 7.

2. Literature review

To develop the optimal distribution system, we first review the literature. Since the customer base in our research is large, we need to identify solution approach that can solve large-scale distribution problems in real-time with best results.

The algorithms for vehicle routing and vehicle scheduling are divided into exact algorithms and approximation algorithms (Dantzig & Ramser, 1959). Exact algorithms include branch and cut (Archetti, Bianchessi, & Speranza, 2014), branch and bound (Almoustafa, Hanafi, & Mladenov, 2013), cutting plane (Christofides, Mingozi, & Toth, 1981), cut-and-column based methods (Contardo, Cordeau, & Gendron, 2014), and asymmetric distance-constrained algorithms (Almoustafa et al., 2013).

The complexity of the exact algorithms grows exponentially with the scale of the problem, hence they are impracticable for large-scale problems. Approximation algorithms are widely used in practice as distribution problems often involve large numbers of customers. Approximation heuristics include traditional heuristic algorithms and advanced intelligent algorithms. Traditional heuristic algorithms include the insertion method (Clarke & Wright, 1964; Gendreau, Hertz, Laporte, & Stan, 1998; Salhi & Nagy, 1999), the exchange method (Potvin & Rousseau, 1993a; Toth & Vigo, 1999), and the interactive optimization algorithm. Advanced intelligent algorithms such as artificial neural networks, genetic algorithms, simulated annealing, and tabu search were later developed (Badeau, Guertin, Gendreau, Potvin, & Taillard, 1997; Brand, 2004; Cordeau & Maischberger, 2012; Fu, Eglese, & Li, 2005). Comparisons of the algorithms are given in Table 1.

Traditional vehicle scheduling algorithms have limitations in application: 1) Euclidean distances are used, which may be infeasible in actual delivery, as restrictions such as one-way streets, road construction, and restraints on truck height/weight may exist; 2) most of the existing algorithms are applied to small-scale problems in which customer size does not exceed $10^3$. In the milk, beverage and snack delivery industries, $10^3$–$10^4$ customers are typical; in a densely populated city, the number is larger.

For large-scale problems, only heuristics (Lin, Yu, & Lu, 2011; Wang & Lu, 2009; Yurtkuran & Emel, 2010) are used to obtain solutions. Few researchers have studied very large-scale vehicle routing problems (VRPs) (Cheong, 2002; Du & He, 2012; Harrison, 1986; Li, Golden, & Wasil, 2005). Previous studies proposed two fundamental approaches. The first is Route-First-Cluster-Second, which means that before scheduling the vehicle route, one would view all customers as points of a long delivery route; then, the large sequence is divided into many small parts based on available resources. Several algorithms have emerged for this setting, e.g. the set covering algorithm (Bramel & Simchi-Levi, 1997; Cullen, 1981), the shortest path approach (Lee, Epelman, White III, & Bozer, 2006), and the set partitioning algorithm (Bowerman, Calamai, & Brent Hall, 1994). However, these methods are often time-consuming with no guaranteed results. The second approach is Cluster-First-Route-Second (Alvarenga & Mateus, 2004; Potvin & Rousseau, 1993a, 1993b; Salhi & Nagy, 1999; Toth & Vigo, 1999) which means before scheduling the delivery sequence, one divides the area into many small regions. It is more practical for large-scale VRPs, thus we choose this approach. Table 2 contrasts the vehicle routing methods in the literature and that of ours.

3. System design

In this section, we first describe the goals of the proposed system. Sections 3.1–3.3 detail the three functional areas. The environment in which the system is developed is discussed in Section 3.4. Section 3.5 presents the system development process.

The large-scale FMCG distributor under study needs to deliver packages to nearly 10,000 customers with quick response times. Customer service is key to business success, and management believes improving the distribution network will strengthen customer relationships. Hence, management is eager to revamp the FMCG distribution system by integrating market analysis, CRM, strategic channel management, and distribution management with real-time inputs from GIS.

Fig. 3 shows that the transactions start with customers ordering items on the Internet. The system then responds by allocating and scheduling resources using customer and resource information and intelligent heuristics. This is the core function of the system. Based on the scheduled distribution plan, the automated warehousing system completes order picking. Finally, vehicles are deployed to deliver orders and serve customers. The system is grouped into four functional areas to support the distribution decision making:

1. The distribution resource management function, which includes information of vehicles, deliverymen, drivers, and customers; distribution history; marketing information; and GIS data (see Section 3.1).
2. The large-scale intelligent distribution decision support function, which supports distribution area clustering and vehicle routing based on GIS and resources available (see Section 3.2).
3. Real time distribution scheduling, which is based on interaction with handheld devices to allow command scheduling and monitoring and tracking (see Section 3.3).
4. A unified platform for marketing, management, and distribution integration. We do not give a separate section for this function, as it simply involves making use of the results from the three functions above for various purposes.

3.1. The distribution resource management function

We now detail the design of the resource management function (see Fig. 4).
Table 1
Comparison of the exact and the approximation algorithms.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact algorithms</td>
<td>• Obtain optimal solutions for small-scale problems</td>
<td>• Computational time too long for complex problems</td>
</tr>
<tr>
<td>Saving algorithm</td>
<td>• Simple with good expansion capability</td>
<td>• Based on Euclidean distance</td>
</tr>
<tr>
<td>Nearest neighbor algorithm</td>
<td>• Simple</td>
<td>• Natural premature convergence</td>
</tr>
<tr>
<td></td>
<td>• Obtain the initial solution within an adjacent table in a relatively short period</td>
<td>• Easy to fall into a local optimum</td>
</tr>
<tr>
<td></td>
<td>• Result of the network is relatively dense</td>
<td></td>
</tr>
<tr>
<td>Heuristic</td>
<td>Basic algorithm</td>
<td></td>
</tr>
<tr>
<td>Nearest inserted algorithm</td>
<td>• Simple using the adjacent table search</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Obtain the initial solution in a relatively short period</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• More accurate than the nearest neighbor algorithm</td>
<td></td>
</tr>
<tr>
<td>Scanning method</td>
<td>• Simple in principle</td>
<td>• A route between two points may not exist in practice</td>
</tr>
<tr>
<td></td>
<td>• Group first then order heuristic algorithm</td>
<td>• Significant differences in the intensity of the workload between different generated distribution routes</td>
</tr>
<tr>
<td>Improved algorithm</td>
<td>Simulated annealing algorithm</td>
<td>• For small and medium-sized problems, a quasi-optimal solution can be obtained</td>
</tr>
<tr>
<td></td>
<td>• For small and medium-sized problems, a quasi-optimal solution can be obtained</td>
<td>• Complex neighborhood conversion and solving strategy</td>
</tr>
</tbody>
</table>

Table 2
Comparison of methods in the literature with ours.

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of customers</th>
<th>Strength</th>
<th>Weakness</th>
<th>Applicability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Archetti et al. (2014)</td>
<td>Branch and cut algorithms</td>
<td>40–60</td>
<td>Exact</td>
<td>Cannot solve large-scale problems</td>
</tr>
<tr>
<td>Almoustafa et al. (2013)</td>
<td>New branch and bound algorithm</td>
<td>up to 1000</td>
<td>Exact, fast</td>
<td>Memory consuming</td>
</tr>
<tr>
<td>Contardo et al. (2014)</td>
<td>Cut and column</td>
<td>12–85</td>
<td>Exact</td>
<td>Cannot solve large-scale problem</td>
</tr>
<tr>
<td>Cordeau and Maischberger (2012)</td>
<td>Tabu search</td>
<td>128</td>
<td>Balance between solution quality and computing time</td>
<td>Many iterations and average result</td>
</tr>
<tr>
<td>Potvin and Rousseau (1993a)</td>
<td>Insertion method</td>
<td>100</td>
<td>Reasonable computing time</td>
<td>Limited type of problem</td>
</tr>
<tr>
<td>Brand (2004)</td>
<td>Modified Tabu search algorithm</td>
<td>50–200</td>
<td>Multi-start strategy</td>
<td>Fast iterations only for small-scale problems</td>
</tr>
<tr>
<td>Lin et al. (2011)</td>
<td>Simulated annealing heuristic</td>
<td>100</td>
<td>Capable of consistently producing quality solutions</td>
<td>Influenced by parameters and limited feasibility</td>
</tr>
<tr>
<td>Du and He (2012)</td>
<td>Nearest neighbor search</td>
<td>6772</td>
<td>Effective for large-scale problems</td>
<td>Euclidean distance</td>
</tr>
<tr>
<td>The proposed system</td>
<td>Genetic algorithm and simulated annealing</td>
<td>9494</td>
<td>Fast; considers workload balance and actual path length</td>
<td>More problems may be studied</td>
</tr>
</tbody>
</table>

The functions are clustered into three subsystems: customer management, deliverymen and driver management, and distribution route management. The outputs here are the basis for large-scale distribution decision-making.

The GIS-based resource management system includes an electronic map, which accepts user requests to filter the database.

Users can choose to display district type (business district, university area, residential area, tourist area, etc.) and mark with different colors. The selections are made by the market type (urban or suburban), customer type (supermarkets, convenience stores, specialty retailers, etc.), customer category (importance level), and service times; the system then prompts the corresponding store...
location on the map. Additionally, regional managers can also extract real-time information of all stores he manages regarding logistics performance and adjust the operations and marketing decisions responsively. This helps coordinate and manage all stores efficiently. Furthermore, users can sort by product criteria such as the sale of a particular brand. The system’s display is also customizable and can show historical sales to a customer, such as the average price and order information.

The CRM function involves query and display of information, including order reporting and location information of customers. We can query customer information, and display the point information. The customer workload and work plan can be optimized for market managers. The system is able to display the sales data and historical data. Through information systems, we can classify customers according to operating status (such as suspension of business, out of business, etc.) and other basic information. The system can display the associated customer service and daily management information. This enables early warning of market anomalies.

The integrated resource management system helps continuous improvement of customer service. The service quality, work efficiency, and customer and supplier satisfaction can thus be improved.

3.2. The large-scale intelligent distribution decision support function

This is the core function of the system. It is very complex and designed to support decision-making for distribution. We borrow the Cluster-First-Route-Second (Potvin & Rousseau, 1993a) approach, and develop a decision support system with the aim to optimize the large-scale distribution.

We improve the clustering method by proposing a new workload measure and balancing method, and apply a hybrid heuristic to route vehicles. The flow of the decision-making process is shown in Fig. 5. The process inside the dashed box runs at the beginning of each planning horizon for partitioning the distribution region. The clusters remain unchanged as long as there are no significant changes in the number of customers, customer location, or fluctuations in customer demand. Afterwards, we focus on VRP (see the process outside the dashed-box). The model and algorithm will be detailed in Sections 4 and 5.

3.3. Real-time management and distribution scheduling

The real-time management and distribution scheduling uses GIS data and GPS technology. In constructing the system, we employ GPS vehicle navigation technology to monitor vehicle location, dispatch vehicles according to road and traffic conditions, and allow remote management, all in real-time (Fig. 6). It improves vehicle operational efficiency and reduces vehicle management costs. In addition, the recorded data can be used to plan marketing activities and employee performance.

The command function allows workers to use a hand-held terminal device to download the order form and navigation routes from the system. The handheld terminal system uses GPS for vehicle routing navigation and gives the current location of the vehicle. In case of emergency, the driver can use the alarm function to alert others to avoid risks. Finally, the driving records can be used to manage customer complaints, traffic accidents, and theft.

The monitoring function refers to the hand-held terminal device’s capability to constantly refresh position and return information. GPS can be used for real-time monitoring of vehicle location. Near field communication (NFC) cards are used to confirm arrivals so managers can check the progress of deliveries. At the same time, the location information will be collected and compared with the database client location in case the position of customers changes.
The purpose of scheduling is to commit to a delivery time through implementing vehicle command and dispatching, and continuously improve customer service levels. Scheduled delivery times are used to ensure quick customer service, real-time distribution, and customer notification prior to arrival. To maintain high quality customer service, the company aims to deliver within 15 min of the scheduled time.

With this system, decision makers in a distribution center can partition the distribution area according to the location of customers and historical demand. Decisions include assigning customers to distribution sub-region and allocating vehicles to routes. The premise is that customer service is on time and workload is balanced. Furthermore, the real-time database is used to determine the delivery route in each distribution area and ensure cost minimization.

The three functions described above (Sections 3.1–3.3) are the base for constructing a platform to integrate marketing, customer service, and managerial decision-making. The platform strengthens logistics process management, reduces costs, and enhances financial performance through better customer service.

3.4. System development environment

The entire system is implemented on the PC using client/server architecture for ease of use, maintenance, and management. Seamless connectivity is achieved with database platforms such as SQL Server, Oracle, or DB2. The WebGIS platform uses SuperMap IS Java, and the mobile terminals are developed on the Android platform, which can be deployed on GPS terminals, smart phones, and PDAs.

3.5. System development process

The system design and development process is summarized in Fig. 7 and further detailed below.
(1) From the geospatial data and logistics information collected, we construct an electronic map of the logistics and distribution region.

(2) We establish the logistics center layer, the customer layer, and the road layer on the map, and each layer contains basic properties such as paths, client nodes, and client data.

(3) We design the distribution network system. We use the Dijkstra algorithm to calculate the actual path of the shortest distance between any two retail outlets in the region, and the derived results are stored in the database.

(4) We develop an integrated logistics resources information management module.

(5) We establish a distribution zoning module, using an improved two-stage K-means clustering algorithm. Vehicle capacity and workload balancing are considered. Distribution area constraints are taken into consideration.

(6) We effectively convert the large-scale multi-vehicle VRP into multiple TSPs. Using a hybrid of the genetic algorithm and simulated annealing, we solve the problem and generate an approximate optimal routing sequence.

(7) We implement a dispatching management module for the vehicles.

(8) We integrate the marketing management, CRM, and logistics management functions.

4. Partitioning the distribution region

In this section, we use the FMCG firm to illustrate the partitioning approach. We first discuss workload measurement (Section 4.1). Thereafter, we present the case under study (Section 4.2), and detail the GIS-based network (Section 4.3). Finally, we propose the clustering method (Section 4.4) and enhance the K-means algorithm to balance workloads among routes. Suboptimal scheduling (imbalance workload) is perceived unfair, as some drivers are assigned to a comfortable and shorter assignment, while others face a cumbersome and long trip, triggering employee conflicts. It is thus crucial to balance workload among routes and among distribution areas.

The literature on distribution workload measurement is surprisingly sparse. Chen and Jiang (2004) take distance, delivery volume, and number of customers as indicators and give them different weights to represent "generalized workload" in the distribution and denoting the workload as:

\[ W_i = K_i m_1 + Y_i m_2 + X_i m_3 \]  \hspace{1cm} (1)

where \( K_i \), \( Y_i \), and \( X_i \) denote the path length of a tour; \( m_1 \) denotes the average delivery volume; \( m_2 \) denotes the number of retail outlets; \( m_3 \) are the respective weights with \( m_1 + m_2 + m_3 = 1 \). However, it is difficult to practically determine the values of the weights \( m_1, m_2, m_3 \). In this research, we propose a new indicator to ascertain the workload. In distribution work, typically the heavier workload, the more working hours are required.

In our case, distribution includes travel and delivery service. The workload can be measured by working hours, which is a function of distance, number of customers, and delivery volume. We can divide the working hours of the distribution process into travel time and delivery service time (see Fig. 8). Through statistical analysis, we find that trip mileage and delivery volume are highly correlated with the work time. Thus, we define the total time as:

\[ T_{\text{total}} = T_s + T_g \]  \hspace{1cm} (2)

where \( T_s \) denotes total service time, and \( T_g \) denotes total travel time.

Details of the derivation of service time and travel time are presented next through the case firm under study.

4.2. The case study

The distribution center of the FMCG in Baotou City, the largest industrial city in Inner Mongolia, currently employs 50 distribution routes. There are 32 vehicles of different types and capacities. Among the 9494 retail outlets, more than 1600 retailers need to be
replenished with inventory daily, and at least one delivery is made to each customer per week.

We track the travel miles, travel time, delivery volume, and distribution points of different delivery routes (see Table 3) and record the problems encountered during the delivery trip. Data collected include the starting and arrival times for each route, and the service duration at each retail site. Table 4 details the daily vehicle tracking record for one route.

### 4.3.1. Study on the service time

Service time is the duration from the arrival of the vehicle at a customer site until leaving the site, where staff completes the removal, clearing, signature, confirmation and other distribution services. After analyzing the service time data, we found that customer service time is weakly correlated with order size, while the association between client class (electronic clearing or non-electronic clearing) and service time is strong, since delivery staff must validate the authenticity of cash/checks.

Fig. 9 shows the frequency distribution of service times, with two distinct peaks, which correspond to the electronic (1st peak) and non-electronic (2nd peak) customers. Thus, service time is greatly affected by payment method, according to which we group the data. We find that the service times of each group are normally distributed, with \( \mu = 1.15 \) and \( \sigma = 0.432 \) minutes for the electronic and non-electronic methods, respectively.

Thus, the expected total service time can be expressed as

\[
T_s = 1.15N_1 + 4.28N_2
\]

in this case study. Where \( N_1 \) denotes the number of electronic settlement customers, while \( N_2 \) is the number of non-electronic clearing customers.

### 4.3.2. Study on the traveling time

Travel time and distance are highly correlated with \( r = 0.921 \). However, heteroscedasticity exists (see Fig. 10), indicating travel time is not normally distributed. Thus, we use Spearman’s rank correlation to test the relationship between distance and travel time and find them to be statistically significant with \( r_s = 0.621 \).

Through field observation, we find that vehicle speed varies considerably. Specifically, vehicles travel faster on highways than in downtown areas because the speed limits are 45.5 and 20 km/h for highway and city driving, respectively. Thus, it is necessary to differentiate distances based on location. We define the distances from the hub to the first customer and from the last customer to the hub as full-speed zones, denoted as \( D_1 \). On the other hand, the distances traveled between customers are within the city and are specified as \( D_2 \). We fit an ordinary least squares (OLS) regression model to estimate full-speed travel time as a function of the distance traveled on highways: 

\[
T_1 = 0.203 + 0.022 \times D_1
\]

where \( D_1 \) is the distance traveled on highways.

---

<table>
<thead>
<tr>
<th>Route no.</th>
<th>Vehicle no.</th>
<th>Time period</th>
<th>Tour mileage</th>
<th>Quantity</th>
<th>Amount ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>bt01</td>
<td>10:44:20–13:11:07</td>
<td>19.72</td>
<td>2361</td>
<td>7728.9</td>
</tr>
<tr>
<td>2</td>
<td>bt04</td>
<td>10:20:30–13:21:40</td>
<td>12.6</td>
<td>2134</td>
<td>7033.5</td>
</tr>
<tr>
<td>3</td>
<td>bt03</td>
<td>9:33:29–12:19:11</td>
<td>27.6</td>
<td>2206</td>
<td>7196.7</td>
</tr>
<tr>
<td>4</td>
<td>bt07</td>
<td>9:05:00–11:32:53</td>
<td>17.2</td>
<td>1903</td>
<td>6121.9</td>
</tr>
<tr>
<td>5</td>
<td>bt15</td>
<td>11:13:13–13:07:15</td>
<td>15.7</td>
<td>3190</td>
<td>10394.5</td>
</tr>
<tr>
<td>6</td>
<td>bt12</td>
<td>9:51:19–12:01:23</td>
<td>11.1</td>
<td>1808</td>
<td>3596.3</td>
</tr>
<tr>
<td>7</td>
<td>bt09</td>
<td>9:56:57–12:38:02</td>
<td>20.7</td>
<td>2567</td>
<td>8336.9</td>
</tr>
<tr>
<td>8</td>
<td>bt02</td>
<td>10:35:30–12:32:47</td>
<td>10.9</td>
<td>1898</td>
<td>6661.0</td>
</tr>
<tr>
<td>9</td>
<td>bt05</td>
<td>9:58:32–12:03:04</td>
<td>12.6</td>
<td>1576</td>
<td>6514.2</td>
</tr>
<tr>
<td>10</td>
<td>bt13</td>
<td>9:40:32–12:24:41</td>
<td>31.8</td>
<td>1232</td>
<td>4024.0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>33</td>
<td>bt14</td>
<td>9:55:00–11:44:44</td>
<td>21.1</td>
<td>1049</td>
<td>7467.9</td>
</tr>
<tr>
<td>34</td>
<td>bt06</td>
<td>9:44:21–11:21:55</td>
<td>5.4</td>
<td>1274</td>
<td>10256.3</td>
</tr>
<tr>
<td>35</td>
<td>bt11</td>
<td>9:06:28–10:38:42</td>
<td>9.5</td>
<td>1108</td>
<td>7433.1</td>
</tr>
</tbody>
</table>

### Table 3

Examples of vehicle tracking record.

### Table 4

<table>
<thead>
<tr>
<th>Customer</th>
<th>Quantity</th>
<th>Amount</th>
<th>Arrival time</th>
<th>Departure time</th>
<th>Distance (km)</th>
<th>Latitude (N)</th>
<th>Longitude [E]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>N40 38.199</td>
<td>E109 48.230</td>
</tr>
<tr>
<td>2</td>
<td>123</td>
<td>2505</td>
<td>10:55:57</td>
<td>10:59:00</td>
<td>4.28</td>
<td>N40 38.199</td>
<td>E109 48.230</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
denotes the highway travel distance in kilometers, and \( T_q \) is the full-speed travel time in hours. Similarly, the city travel time is estimated as: \( T_p = 0.010 + 0.050 \times D_2 \), where \( D_2 \) is the city travel distance in kilometers, and \( T_p \) is the travel time under the city speed limit. Combining the two previous equations, \( T_g = T_q + T_p \) represents the total travel time.

In short, service time is a function of payment method (non-electronic vs. electronic), while travel time is a function of distance and area (i.e. highway vs. city) travelled. We can thus estimate the total travel time (workload) for the route in hours as:

\[
T_{\text{area}} = N_1 \cdot 0.0195 + N_2 \cdot 0.0713 + 0.023 + 0.022 \cdot D_1 + 0.01 + 0.05 \cdot D_2
\]

(3)

4.3. GIS-based distance between locations

There are often various paths available between any two locations. We need to find the best path to ensure the lowest total distance, cost, or time of the route. Finding the best path between two GIS-based points depends on the criterion chosen. For example, to minimize cost, one needs to consider the toll charge and distance traveled, while to minimize the travel time, one has to take the road construction, one-way restriction, congestion, and traffic accidents into account. The distance-based route is the real road distance, not a non-existent straight line computed from the conventional Manhattan distance (Sugano, 1982).

We use \( G = [C', E, V, W] \) to represent the network diagram, in which \( C' = [0, 1, 2, \ldots, n] \) is the set of customer sites, with 0 representing the distribution center. \( E = [(i, j)]; j = 0, 1, 2, \ldots, n, i \neq j \) is the set of edges (roads); \( V \) is the set of path intersections; and \( W \) is a set of positive real numbers such that \( (W: E \rightarrow \mathbb{R}^+) \) denotes the weights corresponding to each edge. Let \( C = \{c_j|(i, j) \in A\} \) be the distance matrix, and \( c_{ij} \) denote the length of segment \((i, j)\). The points sequence \( c_{i_1}', c_{i_2}', \ldots, c_{i_k}' \) is the path from \( c_{i_1}' \) to \( c_{i_k}' \) in G. Among all the paths between \( c_{i_1}' \) and \( c_{j_1}' \), the path with the smallest \( W(c_{i_1}', c_{j_1}') \) is the shortest path. The shortest path

---

**Fig. 9.** (a) Service time distribution. (b) Distribution of electronic payment service times. (c) Distribution of non-electronic payment service times.
between any two retail outlets is calculated by the Dijkstra algorithm (1959).

To establish a matrix (database) for the distances between any two points in the network, we employ the following steps:

(i) Define two sets of vertices $S$ and $T$. $S$ is the collection of all the vertices in which the shortest paths to other vertices in the network have been calculated, while $T = C' - S$ is the set of vertices in which the shortest paths have not been calculated.

(ii) In the initial state, $S$ only includes $c'_0$ (the distribution center). Then, we select $c'_1$ from $T$ such that $c'_1$ has the shortest distance to $c'_0$.

(iii) Each time a new vertex $c'_j$ is added to $S$, the shortest distances from $c'_0$ to the other vertices in $T$ need to be recalculated.

(iv) A new distance between $c'_0$ and a vertex in $T$ will be compared to the shortest distance between the two vertices, and the minimum will be retained as the shortest distance. The new distance is the sum of the shortest path length from $c'_j$ to the vertex under consideration and the shortest distance from vertex $c'_0$ to $c'_j$.

Through the above steps, we are able to find the shortest paths between all nodes.

4.4. The proposed clustering method

The large-scale distribution system involves thousands of customers who are densely distributed in the area. The demand of each customer is much less than the vehicle capacity. To avoid NP-complete and to solve the problem within reasonable time, we partition the distribution area first. In the partition process, we cluster the retail outlets (thousands) into several groups (usually less than 100 retail outlets per group), and schedule the route for each group. The large-scale multi-vehicle, multi-depot VRP is transformed into many single-vehicle multi-depot VRPs. (Fig. 11)

Partition approaches affect the optimization result greatly. It is thus critical to employ a fitting partition method. The following factors are taken into consideration:

(1) Number of clusters. The number of clusters is determined beforehand, and it can be determined by the capacity of vehicles and the total demand.

(2) Customers close in proximity to each other more likely belong to the same cluster.

(3) The workload of the clusters should be balanced or comparable to avoid individual vehicles being overloaded or extremely underloaded.

(4) The clusters are mutually exclusive and collectively exhaustive, i.e. no overlap while covering all customers.

An improved clustering method is proposed in Section 4.5. To ensure fair treatment of deliverymen and drivers, we balance the workload among routes. The traditional K-means method is improved by considering the capacity of each vehicle and the operational time of each driver. Fig. 12 details the proposed algorithm. The goal of the process is to ensure that the work time of each route is comparable.

4.4.1. Determining the value of $k$

In order to determine the number of clusters for the improved K-means algorithm, we use $k = Q/q$, where $Q$ is the total number of items need to be delivered at the time of scheduling and $q$ is the vehicle capacity (in number of items). The firm has $m$ types of vehicles, with $n_1, n_2, \ldots, n_m$ number of vehicles respectively. The daily vehicle costs can thus be minimized by the following model:

$$\min Z = \sum_{i=1}^{m} c_i x_i$$

subject to

$$\sum_{i=1}^{m} v_i x_i \geq Q$$

$$x_i \leq n_i, i = 1, 2, \ldots, m; x_i \geq 0$$

where

$x_i$: is the number of type $i$ vehicles dispatched;

$c_i$: is the cost of dispatching a type $i$ vehicle, i.e. fuel and personnel costs;

$v_i$: is the capacity of a type $i$ vehicle.

By solving Eq. (4), we can find the optimal solution and derive $q$ as follows:

$$q = \frac{v_1 x_1 + v_2 x_2 + \ldots + v_m x_m}{x_1 + x_2 + \ldots + x_m} = \frac{\sum_{i=1}^{m} v_i x_i}{\sum_{i=1}^{m} x_i}$$

Thus, the number of regions (routes) in the network can be determined by:

$$k = \frac{Q}{q} \cdot \frac{\sum_{i=1}^{m} x_i}{\sum_{i=1}^{m} v_i x_i}$$

4.4.2. The initial clustering center

In the beginning, we divide the network into $k$ equal areas. The center points of each region are selected as primary centers of each cluster. In this way, the workload between each of the nodes are balanced, meaning that the clustering will not result in uneven distribution, or unfair treatment among deliverymen and drivers.

4.4.3. Ensuring workload balance

Recall that we use the total distribution time to determine the workload of each route: $T_{\text{work}} = N_1 \cdot 0.0195 + N_2 \cdot 0.0713 + 0.023 + 0.022 \cdot D_1 + 0.01 + 0.05 \cdot D_2$ (see Section 4.2). Let $T_j$ be the total
time of route, \( T_n \) be the minimum route time and \( T_m \) be the maximum route time of the whole network. To balance the workload in the network, we need to control the difference between \( T_p \) and \( T_m \), i.e. \( ||T_m - T_p|| \leq \varepsilon \), where \( \varepsilon \) is a small number designated by management, which is the acceptable deviation among different routes and a constraint to be satisfied in Fig. 12. If \( ||T_m - T_p|| \leq \varepsilon \) cannot be satisfied, then an adjustment is needed.

The method of adjustment is as follows: (i) select the farthest point away from the center of the \( T_m \) group, remove it, and assign it to the group which has the second shortest distance from the removed point. Then mark this point in order to prevent it from being repeatedly chosen and removed in the new group. This step is necessary to prevent the point from forming its own single-point group. (ii) Afterwards, recalculate the workload of these two clusters. Determine \( T_p \) and \( T_m \), and check if \( ||T_m - T_p|| \leq \varepsilon \) again. Repeat until the condition is satisfied.

4.5. The modified K-means algorithm

We now detail the K-mean method we enhanced. It preserves the easiness of implementation, effectively assigns retailers to their suitable cluster, and minimizes the differences in workload among routes for drivers, deliverymen, and vehicles.

Step 1: Select \( k = Q/q \) initial nodes as the cluster centers.

The distribution of initial points is selected by grid. In order to divide the distribution area into \( k \) partitions, we divide the area into \( p \) rows and \( q \) columns such that \( k = pq \). The centers of each partition are chosen as primary nodes.

Step 2: For each cluster center, use the GIS platform to calculate the distances between the center and each of the non-center points. The coordinates of the points are stored as latitude and longitude in a database table where \((x_i, y_i)\) is the coordinate of the center point and \((x, y)\) is the coordinate of point \( i \).

Step 3: For each non-center point, determine the smallest distance to a cluster center, and add the point to the corresponding cluster.

Step 4: Implement the K-means algorithm for clustering. Calculate \( T_j \) for cluster \( j (j = 1, \ldots, k) \) using Eq. (3).

Step 5: Sort \( T_p \) and set \( T_m = \max_q\{T_j\} \) and \( T_n = \min_q\{T_j\} \) where \( m, n \in \{1, 2, \ldots, k\} \).

Step 6: Input tolerance \( \varepsilon \), and set the convergence condition as \( ||T_m - T_p|| \leq \varepsilon \). Stop if the convergence condition is satisfied; otherwise, proceed to step 7.

Step 7: Remove the furthest point in the class of \( m \) and mark this point. The marked point is not to be removed in subsequent iterations. Add this point to the next nearest group (besides group \( m \)). Then recalculate the workloads for these two groups, return to Step 5, and continue to check until convergence.

Management can subjectively specify the maximum workload gap between clusters. For the studied period (a 6-work-day week), there are 6894 retailers under study with total demand equal to 683,973 cases. After applying the proposed partitioning method, we identify 104 routes for the week. In the next section, we optimize the delivery order within each route.

5. The GIS-based routing problem

The GIS-based routing problem is detailed in this section. Section 5.1 proposes an algorithm to solve the individual routing problems, and the algorithm is validated in Section 5.2.

Traditional TSP takes customer sites as nodes and allows vehicles to traverse the network without considering whether an actual road exists between two nodes. The network is an undirected graph as shown in Fig. 13(a). To practically implement the TSP for real-life, it is essential to follow the GIS-based network. GIS often comprises one-way roads, road construction, and height/width restrictions, which invalidate the conventional TSP algorithm. Fig. 13(b) gives a simplified example of a GIS-based network, which shows that direct road connections often do not exist between two customer sites. It includes road intersections, the starting node, the ending node, customer sites, and nearby paths. The GIS database includes the distance, client ID, coordinates, the capacity of each path, etc. Such geographic features supply much richer and realistic information than that assumed by the traditional TSP model.

Fig. 14 gives the network graph-building process necessary to construct a GIS-based traveling path within a cluster (related to
5.1.1. Search area.

Fig. 13. The traditional TSP network vs. the GIS-based network.

Fig. 14. Geographic information build process of GIS-based TSP problems.

Section 4.4 discussion). The GIS platform helps construct an attribute database, which consists of edges (E) and intersections (V) for each cluster. After collecting customers’ latitude and longitude information via handheld terminals, customer information is mapped to the cluster-specific GIS network.

5.1. TSP algorithm based on GIS

From the constructed GIS-based network, we propose a hybrid TSP algorithm to determine the vehicle route for each sub-area. The algorithm is a combination of genetic algorithm (GA) and simulated annealing (SA). It takes advantage of the strong global search ability of GA and the high local search capability of SA. The combination overcomes the deficiencies of each individual algorithm.

5.1.1. The genetic algorithm (GA)

We solve our problems using GA, which contains four components:

a. Encoding scheme.

We use sequence coding, and take the order of traversal as the coding mode, e.g. OABCDEFGH0 means the vehicle travels from the initial node 0 followed by the alphabetical order and finally returns to the initial node.

b. Fitness function.

To ensure the fitness of the VRP encountered is consistent with the conventional concept of fitness in the genetic algorithm, we take the reciprocal of path length as the fitness. The fitness function becomes:

\[ \text{fitness} = K \times \frac{1}{\sum_{i=1}^{n-1} c_{ij}} \]

where \( c_{ij} \) is the length of path \((i, j)\).

\( K \) is a coefficient, selected according to the particular problem.

c. Crossover operator. We set route \(i\) and route \(j\) as a pair of cross routes. This means that another feasible route may be found by combining routes \(i\) and \(j\). The fitness values are \(f_i\) and \(f_j\), where \(\max f\) represents the maximum fitness of all routes. In order to protect good routes, the crossover probability is defined as:

\[ p_c(i, j) = \frac{\max f - \max (f_i, f_j)}{\max f} \]

Since the crossover probability should not be too small or too large, we set the range of crossover probability as \(0.7 \leq P_c \leq 0.9\).

d. Mutation operator. We use the overturn mutation operator, which allows for reordering within a given route. First, we generate two random positions and use the adaptive operator \(P_m\) to adjust according to the fitness value: \(p_m(i) = k_1 \frac{\max f - f_i}{\max f}\), where \(k_1\) is in the range of \((0.01, 0.05)\).

After applying the proposed GA algorithm, we then use the single iteration SA detailed below.

5.1.2. Simulated annealing (SA)

The key elements of SA include the expression of the state, movement, heat balance, and cooling control. At each iteration, SA randomly generates a new route, which will be accepted with an acceptance probability depending on the time-varying “temperature” parameter. It also contains four main components:

(i) Expression of the state.

In SA, a state corresponds to a solution. The energy function of the state corresponds to the objective function of the problem, i.e. the fitness.

(ii) Movement.

For two fitness functions \(f_i\) and \(f_j\), \(\Delta f = f_j - f_i\) represents the objective increment. SA will move from \(i\) to \(j\) when \(\Delta f < 0\).

(iii) Heat balance.

Heat balance is the process of achieving a state of equilibrium at a given “temperature”. This is an internal loop process within SA. We set it to a constant, meaning that we conduct the inner loop iteration the same number of times regardless of the “temperature.”

(iv) Cooling function.

The cooling function is used to control the rate of “temperature” decrease. We select the cooling function \(\text{Temp}_{k+1} = \text{Temp}_k \times r\), where \(r \in (0.95, 0.99)\). Note, the greater the \(r\), the slower the “temperature” decreases, and thus the longer the algorithm runs.

5.1.3. Steps of the hybrid algorithm

Fig. 15 outlines the hybrid algorithm for TSP, which is detailed below:

Step 1: For a cluster with population size \(p\), choose the initial \(K, k_1, \text{Temp}_0\), \(r\), crossover and mutation probabilities, and termination rules. Set the iteration counter to \(n = 0\), and the maximum number of iterations as \(N\).
Step 2: Generate an initial population of routes and calculate the fitness of each. Recall fitness = \( K \times \frac{1}{\sum_{i=1}^{n} c_{ij}} \). To minimize the total distance, we maximize the fitness value in the original population; mark the best route as \( R \) and its corresponding fitness value as \( S \).

Step 3: Implement the genetic algorithm according to the crossover probability and mutation probability.

Step 4: Generate members of the next generation under the replication strategy of the GA.

Step 5: Run SA to derive a new route.

Step 6: Calculate the fitness values for the new routes generated in Steps 4 and 5. Set \( S' = \max f \), and record the corresponding route as \( R' \).

Step 7: Compare \( S' \) and \( S \). If \( S' > S \), then set \( S = S' \), and \( R = R' \). Otherwise, do not update.

Step 8: If \( n = N \), stop; otherwise set \( n = n + 1 \), and return to Step 3.

5.2. Algorithm validation

To validate the proposed hybrid algorithm, we conducted nine simulation tests based on the data provided in TSP lib (http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/index.html) and compared the results with those of the nearest neighbor search algorithm, which is the adjacent secondary algorithm, and the genetic algorithm. The results are given in Table 5, where the values give the average computational times of ten replications. It shows that the hybrid algorithm provides faster results than the nearest neighbor search algorithm and the ordinary genetic algorithm, with 39% and 6% improvements in computational time, respectively.

<table>
<thead>
<tr>
<th>Instance</th>
<th>Nearest neighbor search algorithm</th>
<th>Hybrid algorithm</th>
<th>Genetic algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Att48</td>
<td>65294</td>
<td>35321</td>
<td>37406</td>
</tr>
<tr>
<td>P124</td>
<td>594620</td>
<td>86130</td>
<td>94780</td>
</tr>
<tr>
<td>Swiss42</td>
<td>2172</td>
<td>1390</td>
<td>1463</td>
</tr>
<tr>
<td>Bleep14</td>
<td>38.39</td>
<td>31.2269</td>
<td>31.485</td>
</tr>
<tr>
<td>P107</td>
<td>630710</td>
<td>51070</td>
<td>527827</td>
</tr>
<tr>
<td>Py226</td>
<td>145430</td>
<td>119010</td>
<td>130869</td>
</tr>
<tr>
<td>Py76</td>
<td>478230</td>
<td>122460</td>
<td>135550</td>
</tr>
<tr>
<td>Rat99</td>
<td>1751</td>
<td>1486</td>
<td>1510</td>
</tr>
<tr>
<td>Kroc100</td>
<td>39094</td>
<td>25079</td>
<td>26923</td>
</tr>
</tbody>
</table>

Table 5: Contrast of computational times (in seconds) of different algorithms.

6. Implementation and results

To understand the effectiveness of the proposed system, we examine the costs, efficiency, and customer service improvement after the implementation of the proposed hybrid algorithm in the studied firm (the FMCG distributor). Table 6 contrasts the differences in firm performance before and after system implementa-

tion. We found that the weekly travel time of all vehicles has reduced from 485 to 324 h, a 33% improvement.

Table 7 gives a glimpse of some distribution routes. When the total time (the rightmost column) is less than 5 h, the route will be combined with another nearby route to reach the 8-h workday requirement. Through such combination, we are able to reduce the total fleet requirement from 16 to 11 trucks. In summary,

1. The number of distribution routes decreased from 123 to 104, a total reduction of 19 routes, corresponding to five vehicles.
2. Full-load trips increased from 86 times to 104 times, or from 86% of all trips to 99% of all trips.
3. Delivery time decreased by 161 h.
4. The total mileage decreased by 613 km.

The proposed system helps achieve the goals of the distribution center: cost is minimized while efficiency and quality are improved. The logistics costs of the distribution center fell from $4.75 million to $4.25 million immediately following the implementation of the system. The single-package management costs are $30.31 now compared with China’s national average of $39.57, which is 23.40% less. The single-package distribution costs are $11.53 per box, compared with the national average of $14.93 per box, a 22.77% difference. The logistics costs account for 0.87% of the sales.
Table 6
Results before and after the hybrid algorithm implementation.

<table>
<thead>
<tr>
<th>Performance measure</th>
<th>Before optimization</th>
<th>After optimization</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of distribution areas (number of vehicles)</td>
<td>123 areas (16 vehicles)</td>
<td>104 areas (11 vehicles)</td>
<td>9 areas (5 vehicles)</td>
</tr>
<tr>
<td>Total distribution time</td>
<td>484.78 h</td>
<td>324.05 h</td>
<td>160.73 h</td>
</tr>
<tr>
<td>Total distance</td>
<td>542.3 km</td>
<td>4809.776 km</td>
<td>633.224 km (11.3%)</td>
</tr>
<tr>
<td>Number of full-load trips</td>
<td>86</td>
<td>104</td>
<td>18</td>
</tr>
<tr>
<td>Proportion of full-load trips</td>
<td>70%</td>
<td>~24 h</td>
<td>~24 h</td>
</tr>
<tr>
<td>Customer waiting times</td>
<td>~48 h</td>
<td>~48 h</td>
<td>~48 h</td>
</tr>
<tr>
<td>Yearly distribution costs</td>
<td>$4.75 million</td>
<td>$4.25 million</td>
<td>$0.5 million</td>
</tr>
</tbody>
</table>

Table 7
The optimization results for each route in the studied firm.

<table>
<thead>
<tr>
<th>Distribution zone</th>
<th>Number of retailers</th>
<th>Total deliveries</th>
<th>Mileage (km)</th>
<th>Total time (h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>77</td>
<td>6526</td>
<td>143.833</td>
<td>7.25</td>
</tr>
<tr>
<td>2</td>
<td>71</td>
<td>6554</td>
<td>140.053</td>
<td>7.03</td>
</tr>
<tr>
<td>3</td>
<td>72</td>
<td>6648</td>
<td>69.415</td>
<td>4.18</td>
</tr>
<tr>
<td>4</td>
<td>89</td>
<td>6513</td>
<td>123.154</td>
<td>6.66</td>
</tr>
<tr>
<td>5</td>
<td>65</td>
<td>6509</td>
<td>57.882</td>
<td>3.58</td>
</tr>
<tr>
<td>6</td>
<td>61</td>
<td>6536</td>
<td>56.663</td>
<td>3.46</td>
</tr>
<tr>
<td>7</td>
<td>79</td>
<td>6737</td>
<td>59.192</td>
<td>3.96</td>
</tr>
<tr>
<td>8</td>
<td>81</td>
<td>6742</td>
<td>75.434</td>
<td>4.00</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Total</td>
<td>6894</td>
<td>683,973</td>
<td>4809.776</td>
<td>324.05</td>
</tr>
</tbody>
</table>

revenue, which is 2.25% lower than China's national average of 0.89%.

7. Conclusions
We address the large-scale vehicle planning, scheduling, and routing problem of a distribution network based on GIS technology. The research starts from the study of workload measurement with the aim to find the approximate optimal vehicle routing plan in order to reduce costs, improve work efficiency, and improve customer service. The solution framework proposed in this paper is based on the analysis of a large-scale FMCG distributor in a densely populated city with many customers. The objective function, constraints, and assumptions of the model come from a real distribution process. The proposed solution scheme is to divide and conquer. First, divide the large-scale complex distribution network into several small-sized distribution areas subject to system constraints. The path of the multi-vehicle optimization problem is reduced to a number of TSPs. We design a heuristic algorithm to solve the TSP for each sub-region and then combine the results to obtain a quasi-optimal solution for the entire distribution area.

In partitioning distribution area, we propose an improved two-stage K-means clustering method and divide the area based on workload balance. For vehicle routing in the sub-regions, we propose a heuristic algorithm, which integrates the genetic algorithm and simulated annealing. The hybrid algorithm uses GIS data to reflect the true environment. The proposed method provides a comprehensive solution framework for large-scale distribution problems. The efficiency of the algorithm is simulated and verified. Overall, the performance improvements: cost savings, efficiency enhancement, and high customer satisfaction, have greatly increased the competitive advantage of the FMCG distributor.

Acknowledgments
This research is supported by the National Natural Science Foundation of China (71572011), the fundamental Research Funds for the Central Universities (FRF-BR-15-011B).

References
Salhi, S., & Nagy, G. (1999). A cluster insertion heuristic for single and multiple de-
pot vehicle routing problems with backhauling. *Journal of the Operational Re-
search Society*, 50(10), 1034–1042.
Sugano, O. (1982). The distribution of the random Manhattan distance in a hyper-
vehicle routing problems with backhauls. *European Journal of Operational Re-
search*, 113(3), 528–543.

Wang, C., & Lu, J. (2009). A hybrid genetic algorithm that optimizes capac-
2921–2936.
3427–3433.