Hindawi Publishing Corporation
International Journal of Vehicular Technology
Volume 2012, Article ID 963427, 14 pages
doi:10.1155/2012/963427

*Research Article*

# A Geo-Aware and VRP-Based Public Bicycle Redistribution System

**J. H. Lin[1] and T. C. Chou[2]**

[1] *Department of Electrical Engineering, Nan Kai University of Technology, Nantou 54243, Taiwan*
[2] *Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan*

Correspondence should be addressed to J. H. Lin, jhlin@nkut.edu.tw

Received 24 July 2012; Revised 2 November 2012; Accepted 24 November 2012

Academic Editor: Chyi-Ren Dow



Public Bicycle System (PBS) has been developed for short-distance transportation as a part of the mass transportation system. The supply and demand of bikes in PBS is usually unbalanced at different stations and needs to be continuously and widely monitored and redistributed. The bicycle redistribution is a part of the vehicle routing problem (VRP). We can apply solutions to the VRP to redistribute bicycle efficiently. However, most solutions to the VRP use the Euclidean distance as the condition factor, which does not take road conditions, traffic regulations, and geographical factors into account, resulting in unnecessary waste of delivery time and human resources. In this work, we propose an actual path distance optimization method for the VRP to adapt the several additional constraints of road problems. We also implement a system that integrates real-time station information, Web GIS, the urban road network, and heuristics algorithms for PBS. The system includes a simulator inside that can assist PBS managers to do the route planning efficiently and find the best scheduling strategy to achieve hotspot analysis and the adjustment of station deployment strategies to reduce PBS operation cost.

## 1. Introduction

In recent years, issues on carbon emission and energy saving have been taken seriously. Large cities such as New York, London, Paris, Tokyo, and Singapore are encouraging green commuting instead of using gasoline-powered vehicles to solve traffic and air pollution problems. Therefore, Public Bicycle System (PBS) has been developed for short-distance transportation. As a part of the mass transportation system, public bikes must be highly reliable. The scheduling and distribution determination of the public bikes renting system must operate smoothly.

The supply and demand of bikes is usually unbalanced at different stations [1, 2]. Due to unbalanced travel patterns and topographic effects, some stations will have no bicycle and some stations will be full of bicycles. People cannot rent bicycles at empty stations and return bicycles at full stations. To address this unbalance problem, bicycles need to redistribute from full stations to empty stations via specially designed trucks.

The bicycle redistribution is a part of the vehicle routing problem (VRP). We can apply solutions to the VRP to redistribute bicycles efficiently. There are some solutions to the VRP [3–5]. However, these solutions have some problems. Most studies for the VRP were focused on standard conditions, which were not suitable for the specific cases. For example, many studies assume that destinations are on the two-dimensional plane and they briefly use the Euclidean distance [6] to calculate the transportation cost. However, in the real world, it must be reformulated to take into account road conditions, traffic regulations, and geographical factors that result in longer delivery distance and transport time.

The bicycle redistribution is a part of logistics distribution. The logistics distribution is a kind of spatial information activities that we need GIS to support it. Most public bicycle services provide real-time information (displays the cycle hire docking station locations and rental conditions) for both renters and managers on Web GIS. The service allows renters using smart phones to find the nearest station and displays the number of available bicycles and the number of free docking stations in real time.

Current PBSs only provide limited information that they do not provide more detailed information and additional functions for management [7, 8]. The managers need more powerful tools to help them efficiently redistribute bicycles and ongoing monitoring of their bicycles over the course of the day. In some cases, the service operators may add new bicycle rental stations or adjust the existing station location to improve service efficiency. Through a simulator with a powerful and friendly interface, the staff can quickly obtain the possible effects easily. Thus we design and implement a system to help managers make the routing planning and decision-making.

In this work, we propose an actual path distance optimization method for the VRP to adapt the several additional constraints of road problems. On the other hand, our system integrates Google Maps technology, which provides a simple but very powerful map interface. The distribution sites and the optimization of distribution routes were displayed on Google Maps that can assist the PBS managers to archive management tasks.

The rest of this paper was organized as follows. Section 2 describes related work. Section 3 describes our actual path distance optimization method. Section 4 discusses design and the implementation of our system. Section 5 provides the experimental results and the analysis of our scheme. Section 6 shows the conclusions.

## 2. Related Work

In this section, we first describe the bicycle redistribution problem and some applications of the popular public bicycle services in the world. Then, we introduce several variations and the specializations of the VRP. Finally, we describe methods for solving the VRP.

*2.1. Bicycle Redistribution.* There are a variety of technologies and different service types in an existing and proposed bike share program. The vast majority of urban PBS feature is fixed stations. It means that bicycles are locked at designated docks when not in services. This kind of service usually causes user serious problems. As shown in Figure 1, a user cannot rent a bicycle due to an empty station and a user cannot return a bike due to full station. The bike redistribution problem becomes an important issue.

The fixed systems need constant monitoring to ensure bikes are available for picking up and docking off at every station. The stations are connected with an electronic monitoring center via the network. Through the redistribution mechanism and motorized vehicles, the electronic monitoring center rebalances bicycles between empty stations and full stations.

The public bicycle systems in different cities have different characteristics requiring different redistribution mechanisms. For example, Bicing in Barcelona, cyclists generally ride bicycles to the town at downhill but take other transportation to go back uphill since Barcelona downtown is located at the bottom of a bowl shaped valley and people dislike uphill. In Barcelona's case, the operator uses a larger than the usual fleet of redistribution vehicles continuously taking bikes back to uphill stations [9]. Another example in Paris's Vélib, it is a huge density of station of a bike share program that has 20,600 bicycles and 1,451 stations in the city. Paris's Vélib has 23 natural gas powered redistribution vehicles working 24 hours a day/7 days a week to redistribute bicycles. In addition, some small public bicycle programs usually set the safety stock level for each station and irregular redistribution time to reduce redistribution costs.

The operating cost of the bike sharing system is expensive. In Taiwan, YouBike has 500 bikes and 11 service stations in Taipei City results in $NT millions deficits after running one year. It cost $NT three hundred thousand in the C-Bike in Kaohsiung each month. The bicycle redistribution is one of the expensive costs in PBS. The operational cost of Vélib for the redistribution a bicycle is about $3. Service staff in Bicing is about 230 people and 50% of them are assigned to bicycle movement. How to reduce the cost is critical for PBS.

*2.2. The Information Service for the Public Bicycle System.* Many cities around the world have bicycle sharing systems. They provide some applications for iPhone, Android, and Web services that help users to find the nearest docking stations for hiring bikes by GPS. Some of those applications were shown in Figure 2. In some websites, they show real-time information about available bikes and empty slots.

Rongliang and Yunru [7] and Luo et al. [8] designed and implemented a Google Maps-based information system for the public bicycle transportation program in Hangzhou city, China. They use the Google Maps API and AJAX technique to visualize information and to display the locations of the bike rental sites, available bikes, and bike parking docks for renters and management staff.

Several authors proposed the different developments of the applications on Google Maps. These application services categorized as mashup applications [10, 11]. "Mashup" has become one of the hottest buzzwords in the Web applications area. Many companies, organizations, and institutions are rushing to provide mashup solutions or relabeled existing integration solutions as mashup tools, just like these applications [7, 8, 12–17].

For decision support application, Su et al. [16] proposed a Web-based cycling routing planner system that assists cyclists to find their personal cycle routes. The system was designed and implemented with a user friendly interface using the Google Maps. The system allows users to select cycle routes based on users' preferences regarding air quality, safety, total cycling distance, elevation gain, and areas featuring vegetation. Santos et al. [15] designed and implemented a trash collection decision support system based on the Google Maps services. They proposed a spatial decision support system based on the Web aiming to generate optimized trash collection routes for capacitated arc routing problem (CARP) that involves serving the demand, respectively, about a set of arcs on an urban road network. This system integration includes the Google Maps service, a classical heuristic and

(a) Empty station (b) Full station

Figure 1: Empty station and full station.


(a) (b) (c)

Figure 2: iPhone App for Cycle Hire.

ant-colony meta heuristic, and a database system to help the planner to generate detailed vehicle routes on the graphical visualization map. This trash collection decision support system was tested in Coimbra, Portugal.

Kawano et al. [13] proposed a GIS-based solution to the vehicle scheduling and routing problems in a day-care center, which uses two spatial technical tools, “PhotoTrackr” and “ArcGIS,” to record and analyze vehicle routes. The authors use the car tracking as the actual path information. But the car tracking only contains the route information that will be inaccurate since it does not have real-time traffic information. Also, the system is used in a day-care center where the customers’ trait is almost changeless so that it can find the appropriate route for each customer beforehand. In the bicycle redistribution problem, empty and full stations change often. Consequently, the scheme is not suitable for this problem.

*2.3. The Vehicle Routing Problem.* The Traveling Salesman Problem (TSP) [18] and the Vehicle Routing Problem are two of the popular problems in the field of the combinatorial optimization. The VRP is a kind of TSP for derivate problems, and TSP is how to find the shortest path connecting a number of locations which starts and ends at the same depot. The largest difference between the VRP and the TSP is that the VRP must consider the vehicle capacity constraints. If the nodes' total demand surpasses the vehicle load limit, it is necessary to increase the number of vehicles for distribution services.

The VRP is an NP-hard problem that needs heuristic methods to solve it. In [3] proposed by Bräysy and Gendreav in 2005, the basic VRP is concernedwith finding a set of routes to serve the geographically dispersed customers and to minimize the total travel distances. In the VRP, the distribution of vehicles from a single location has to pick up or drop off the goods and then return to the original location. Each vehicle can carry with limited capacity or travel time may be restricted.

Heuristic methods for the VRP are usually classified into 3 subgroups [19]—the construction method, the two-phase method, and the improvement method. First, the construction method gradually builds a feasible solution by choosing minimizing transport cost. It is easy to implement and executes quickly like the nearest neighbor method [20] and the random method do. Next, the two-phase method divides the solution process into 2 parts—clustering of all nodes as feasible travel routes ignoring their sequence and route construction. An example of a two-phase method is sweep algorithm [21]. Third, the improvement method begins with one feasible route and tries to improve tours by exchanging visit sequence for nodes or arcs between the routes. In this case, if a solution is better than the current solution, it becomes a new solution. The procedure is repeated until there is no better solution in the current solution.

The advantage of classical heuristics is that they have polynomial runtime. Thus, using the improvement heuristics method can be a better solution within a reasonable period of time. In addition, these methods only do a limited search to find a local optimum solution in the solution space.

## 3. The Actual Path Distance Optimization Method

Due to unbalanced travel patterns and topographic effects, bicycles need to redistribute from full stations to empty stations via specially designed trucks. How to redistribute bicycles efficiently between empty and full stations is the problem of the VRP.

The VRP definition states that $m$ vehicles initially are located at a depot and deliver discrete quantities of goods to $n$ customers. The solution of the VRP is a set of routes that start and finish at the same depot, and they must satisfy the constraint that each customer is served only once. The goal is to minimize the total transportation cost (e.g., travel distance and travel time).

To apply the problem of the bicycle redistribution in the VRP, the specially designed trucks are the vehicles which have limited capacity to transport bicycle while stations are the places where the trucks need to redistribute bicycles. The goal is to minimize the total cost that the PBS can save more redistribution cost. The following notations are used in the VRP and we transform them for the problem of the bicycle redistribution in Table 1.

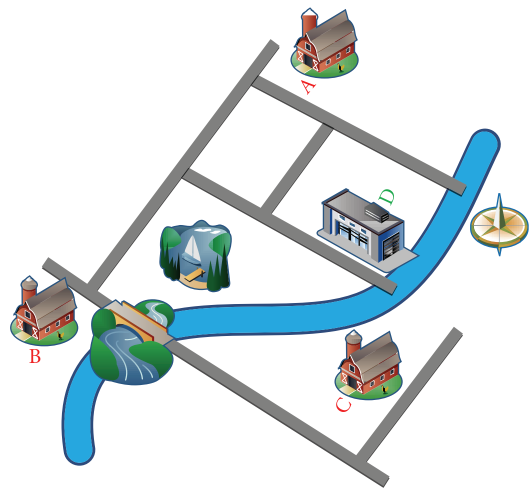



Figure 3: The impact of road infrastructure on tour sequence.

Most studies for solving the VRP estimate the transportation cost between nodes by geographical coordinates since they do not have detailed information. Therefore, they use the straight line distance as the transportation cost by calculating the Euclidean distance [22, 23]. But, the actual path between two nodes on the real world is not a straight line path because of the presence of obstacles or restriction on traffic regulations.

For example, Figure 3 shows the impact of road infrastructure on the tour sequence. Assume there are three stations A, B, C, and one depot D on the road network map. The Euclidean distance between the depot and the three stations is $\overline{DC} < \overline{DA} < \overline{DB}$, and actual path distance is (D, A) < (D, B) < (D, C). If we only consider the Euclidean distance as the TSP decision criteria, the tour sequence will be D → C → A → B → D. $\overline{DC}$ has the shortest distance that we choose station C as the first station to visit. But the Euclidean distance is unreliable in the road network. Vehicle cannot cross the river so we have to detour the river via the bridge to reach station C, which results in longer travel distance. According to the actual path distance, the travel distance of the tour sequence D → A → B → C → D will be shorter than the Euclidean distance.

In Figure 4, we can see the difference between the VRP and the VRP based on the actual path distance. In the real world, there are many intersection nodes on the routes. The VRP based on the actual path distance is essentially the same as the classical VRP. The constraints of the optimization are listed below.

(i) We have to obtain more detailed information for mapping the straight line path to real road path.

(ii) We have to convert the road network from the undirected graph to the directed graph.

(iii) We need extra memory space to save road features.

The method of the classical VRP for searching all of the nodes is relatively simple by using the Euclidean distance which the distances in of go and return path are the same.

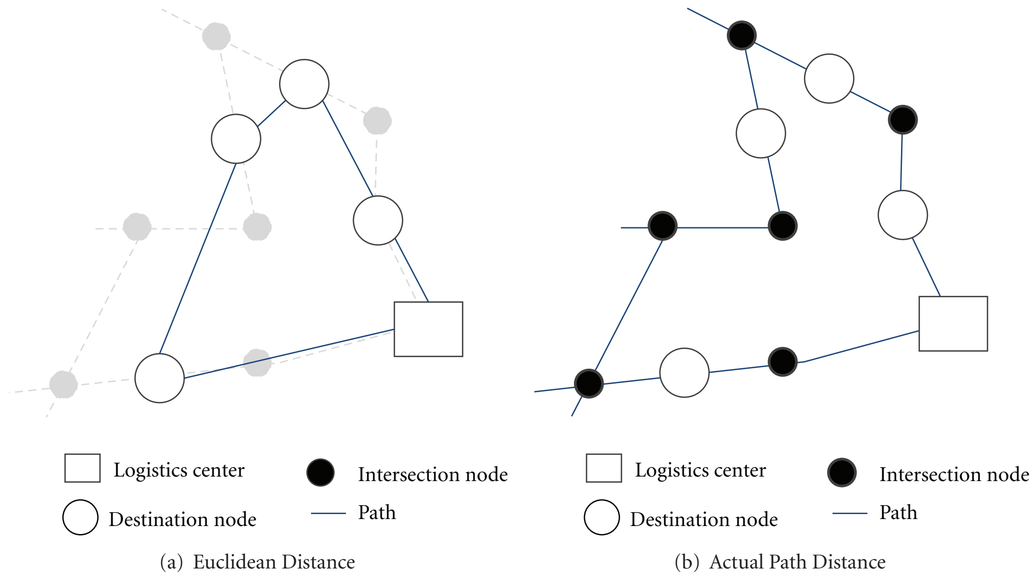



Figure 4: Euclidean distance versus actual path distance.

Table 1: Notations for the VRP and bicycle redistribution.

| Notation | VRP | Bicycle redistribution |
|---|---|---|
| $m$ | The number of vehicles | The number of vehicles |
| $n$ | The number of vertexes | The number of stations |
| $q_i$ | Demand/supply at vertex $i$ | Demand/supply at station $i$ |
| $d_i$ | Service duration at vertex $i$ | The time of redistributing bikes at station $i$ |
| $C_{ij}$ | Cost to traverse arc or edge $(i, j)$ | Travel cost from stations $i$ and $j$ |
| $D_{ij}$ | Distance from vertex $i$ to vertex $j$ | Distance from stations $i$ to $j$ |
| $T_{ij}$ | Travel time from vertex $i$ to vertex $j$ | Travel time from stations $i$ to $j$ |
| $C$ | The capacity of vehicles | The capacity of vehicles |
| $T$ | The maximum route duration of vehicles | The maximum working time of staff |

But the actual path distance method for the VRP uses the road network consisting of the undirected graph constructed by intersection nodes, destination nodes, and paths in which go and return path might be different. The data structure used in our scheme includes not only the path distance and node information, but also the path capacity, one-way/two-way street, and the other road features. To use the actual path distance replacing the Euclidean distance greatly expands the classical VRP network diagram and makes the VRP model meets the real-world problems.

How do we obtain the actual path distance? We can use Google Directions to solve it. Google launched the Google Directions API in May 2010 that allows developers to develop a new service easily into their applications. The Google Directions API is a service that calculates distances, travel time, or navigation services between locations by using an HTTP request. Through the Google Directions API, it can overcome the traffic and roads restrictions such as the maximum of the speed limit, banned turns, and one-way streets. In addition, it provides severable accuracy driving time for streets in each city according to the traffic congestion and the maximum of the speed limit. We use the API to help us obtain the travel distance and estimate travel time between origins and destinations. In addition, the estimated time is very close to the current situation of the road conditions.

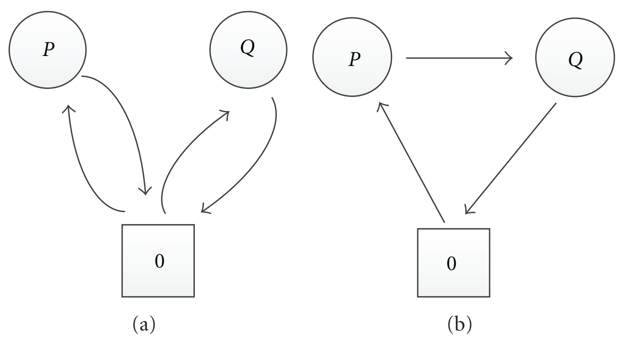



Figure 5: The savings method concept.

*3.1. Classical Heuristics Schemes Based on the Actual Path Distance.* All classical heuristic schemes belong to the construction heuristics or the two-phase methods. We enhance classical heuristics with the actual path distance optimization method by using Google Directions.

*3.1.1. The Nearest Neighbor Algorithm.* The nearest neighbor algorithm is an approximation algorithm to solve the travelling salesman problem. The algorithm only gives a rough attempt to obtain a better tour. It can quickly yield a short tour for the TSP, but the solution is usually not the optimal one. Here are the steps of the algorithm.

*Step 1.* Start on a home node (depot node) as a current node.

*Step 2.* Find out the nearest unvisited node P and add it to current tour.

*Step 3.* Set the current node to P.

*Step 4.* Mark P as visited.

*Step 5.* If all the nodes are visited, then terminate.

*Step 6.* Go to Step 2.

We enhance the algorithm with the actual path distance by using information like distance and traveling time, which were calculated by Google Directions API and the consideration of vehicle capacity to find out the nearest node. Then we set the node as a reference point and repeat the action until all nodes are visited.

*3.1.2. The Savings Method.* In 1964, Clarke and Wright proposed the savings method [24, 25] to solve the VRP. The first step of the savings method is to calculate "savings" for each pair of customers. In Figure 5, there were two customers P and Q. $C_{P0}$ was the transportation cost between the depot and the customer P, while $C_{Q0}$ was the transportation cost between the depot and the customer Q. $C_{P0}$ and $C_{0P}$ were symmetric when the distance was calculated by using the Euclidean distance. In Figure 5(a), the total distance for visiting customers P and Q was $2C_{0P} + 2C_{0Q}$. Figure 5(b) showed that the three targets could feasibly be merged into a single route ($0 \rightarrow P \rightarrow Q \rightarrow 0$) and the distance was $C_{P0} + C_{0Q} + C_{PQ}$.

The total transportation distance $D_a$ in Figure 5(a) was

$$D_a = C_{0P} + C_{P0} + C_{0Q} + C_{Q0}. \tag{1}$$

Equivalently, the transportation distance $D_b$ in Figure 5(b) was

$$D_b = C_{0P} + C_{PQ} + C_{Q0}. \tag{2}$$

By subtracting the two routes, we obtained the savings $S_{PQ}$ :

$$S_{PQ} = D_a - D_b = C_{P0} + C_{0Q} - C_{PQ}. \tag{3}$$

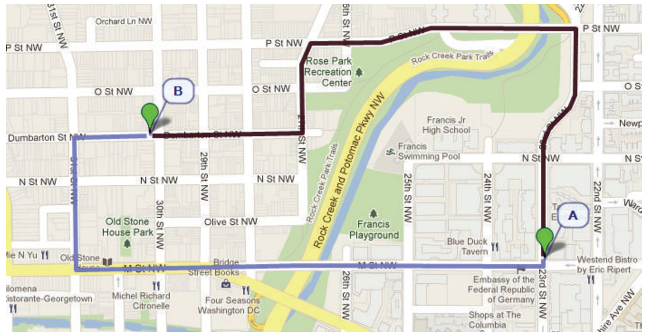



FIGURE 6: The Go and Return Path is Different.

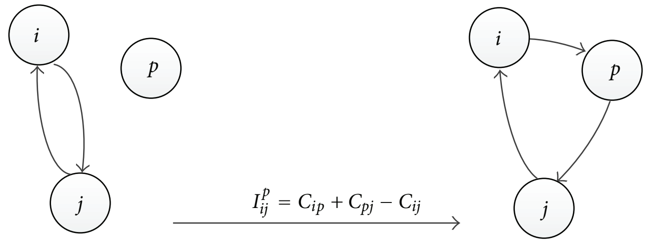



FIGURE 7: Illustration of the insertion concept.

The savings were calculated and ranked. Those nodes were joined into routes which could be linked (otherwise, added a new route) until a constraint (maximum working hours, vehicle capacity) was reached.

In the general savings method with $n$ customers, the go and return path is the same that we need to calculate $n(n-1)/2$ savings. But in the real-street network, the go and return path may be different that $C_{ij} \neq C_{ji}$. The travel costs may be asymmetric that we need to calculate $n(n-1)$ savings.

In Figure 6, there were two nodes (the point A and the point B) on the map. The point A was the starting point and the point B was the destination. When we used Google Directions API to obtain the actual path distance, we could see two different results (the light blue line represented the go path and the dark brown line represented the return path). So that when we used the actual path distance instead of the Euclidean distance to obtain the savings matrix, we must consider this issue.

*3.1.3. The Farthest Insertion Algorithm.* The strategy of the farthest insertion algorithm [26] is to fix the overall layout of the tour early in the insertion process. Its concept is a hard beginning making a good ending. The farthest insertion algorithm works as follows in three steps.

The Initialization Step

(1) Start with a subgraph consisting of a node $i$ only. For example, in TSP, it usually starts at a depot.

(2) Find a node $p$ that is the farthest node from the depot. Join $p$ and the depot together, such that $C_{ip}$ is the maximal distance and form the subtour $i$-$p$-$i$.

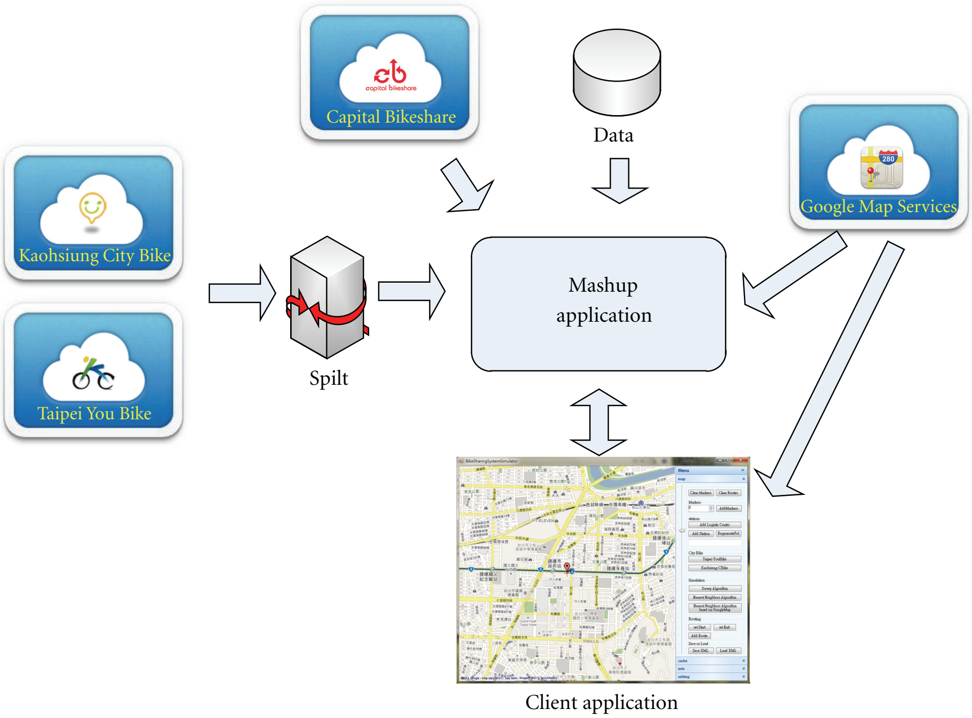



Figure 8: The overview of the system.

The Selection Step

(3) Give the current subtour. Find a node $p$ that is not in the sub-tour and the node $p$ is the farthest node from the edge between the sub-tour.

The Insertion Step

(4) Find the arc$(i,j)$ in the sub-tour, which is the minimum (like savings method $C_{ip} + C_{ip} - C_{ij}$, find the minimum saving value). Insert the node $p$ between $i$ and $j$ as shown in Figure 7.

(5) If all the nodes are added to the tour, then stop. Else go to the Step 3.

In this method, we also applied the actual path distance in the farthest insertion algorithm by using Google Directions. The same issue was discussed in the savings method. We must consider the problem of moving direction, travel costs may not be the same that $I_{ij} \neq I_{ji}$.

## 4. System Design and Implementation

In this section, we designed and implemented the geo-aware and the VRP-based public bicycle redistribution system.

*4.1. System Design.* The overview of our public bicycle redistribution system was shown in Figure 8. It was a typical mashup [27] application that integrated external data source with Google Maps and the third-party websites. We combined station information data from City Bike [28], You Bike [29], and Capital Bikeshare [30] as the data sources and integrated with the Google Maps.

Google Maps is an application of Web GIS. It is a classical Web application service based on AJAX and provides free satellite images. We could use location icons, metadata, and the location coordinate into the Google Maps interface. The Google Maps API supports the development of the spatial data including several services, retrieving static map images, performing geo-coding (converting addresses into geographic coordinates), generating driving directions (directions between two or more points by specifying waypoints), and obtaining elevation profiles.

Mashup is a kind of Web page or application that combines data or functionality from two or more independent components to create a new application, view, or service. In this work, there are 3 stages in the mashup.

(1) Data collection: use the agent program to collect the rental data of the public bicycle sharing systems in Taipei and Kaohsiung cities as material.

(2) Data connection: process the received data, integrate other data, and compute the geographic information.

(3) Data display: display the results on Google Maps.

We have chosen the layered architecture as shown in Figure 9 to design the system. Four-layer services application is presented here, which is a relaxed architecture. The four layers are

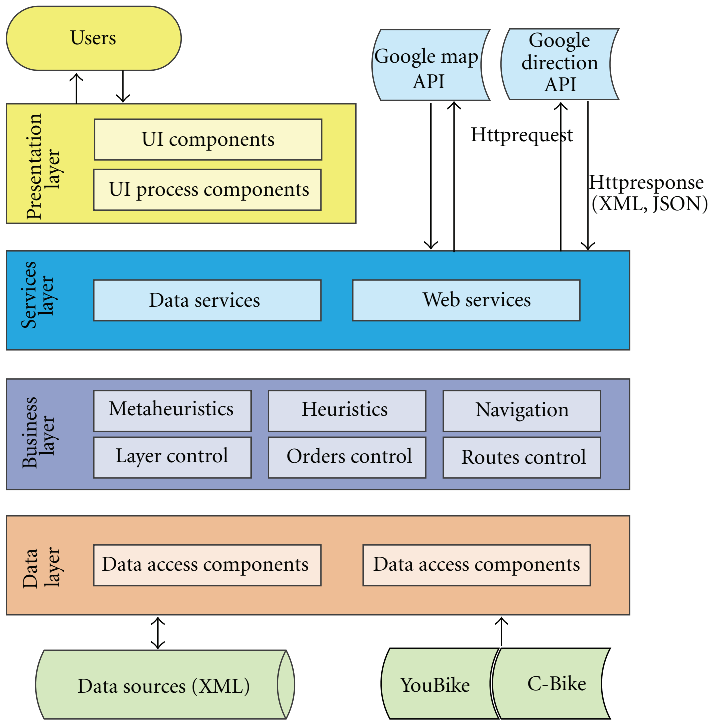



Figure 9: System architecture.

(i) the presentation layer: the presentation layer provides application interface for users. We use "UI components" and "UI process components" to implement Win-Form-based interfaces with. Net technology.

(ii) the services layer: we use Web services as the implementation of technology for SOA, which embeds a map provided by Google Maps and combines its API with the system.

(iii) the business layer: the business layer contains components used to implement business logic. The business components perform the core processing logic within the system. We include the following functionality: logical calculation and routes display logic.

(iv) the data layer: the data layer processes the physical storage and the retrieval of data. For instance, we can load or save station information from an XML file.

This system was implemented and integrated the VRP algorithms and used Google Maps to request the street network and topography data via the Internet. The PBS planner can enter the planning capacity to define the input and output requirements. The system has the ability to find better solution with different VRP algorithms and bicycle redistribution mechanisms with simulation function. This system will output the final route map for individual vehicles, estimated work hours, computing performance, and the comparison analysis of the VRP algorithms.

Simulation was divided into 3 stages, the initial stage, the routing stage, and the analysis stage as shown in Figure 10. In the initial step, we used Google Maps API to produce the street map and the region map. Then we combined the logistics center location with the destination nodes to make a raw network dataset for the routing stage. In the routing stage, we designed four different classes to control the map operation, orders layers class (nodes order management), depots layers class (nodes display control), routes layers class (path management), and navigation class (route guideline and navigation services). We used Google Directions API to obtain the relationship within each node and produced the distance matrix as a secondary dataset. At this stage, we also designed different computing modules for different algorithms and simulation types (e.g., VRP or TSP). The final Stage was to output optimizing vehicle routes with graphical representation on Google Maps and pop up the analysis result.

*4.2. System Implementation.* In this system, the data processing included two parts, the spatial data processing on Google Maps and raw data processing on the third-party websites. In Taipei and Kaohsiung cities, the bike rental system provides information services for users through Web pages as shown in Figure 11. It shows maps of Taipei and Kaohsiung overplayed with small markers or html table of available bicycles and free docks for each PBS station. This real-time information is inserted into the Web pages using html table or the JavaScript code with a string of variables that included rental information for each station. However, it is not a standard XML file that we need to analyze and spilt this JavaScript file when computing the VRP. We also implement the standard Web service to obtain station information from Capital Bikeshare in Washington, D.C., which provides a standard XML file.

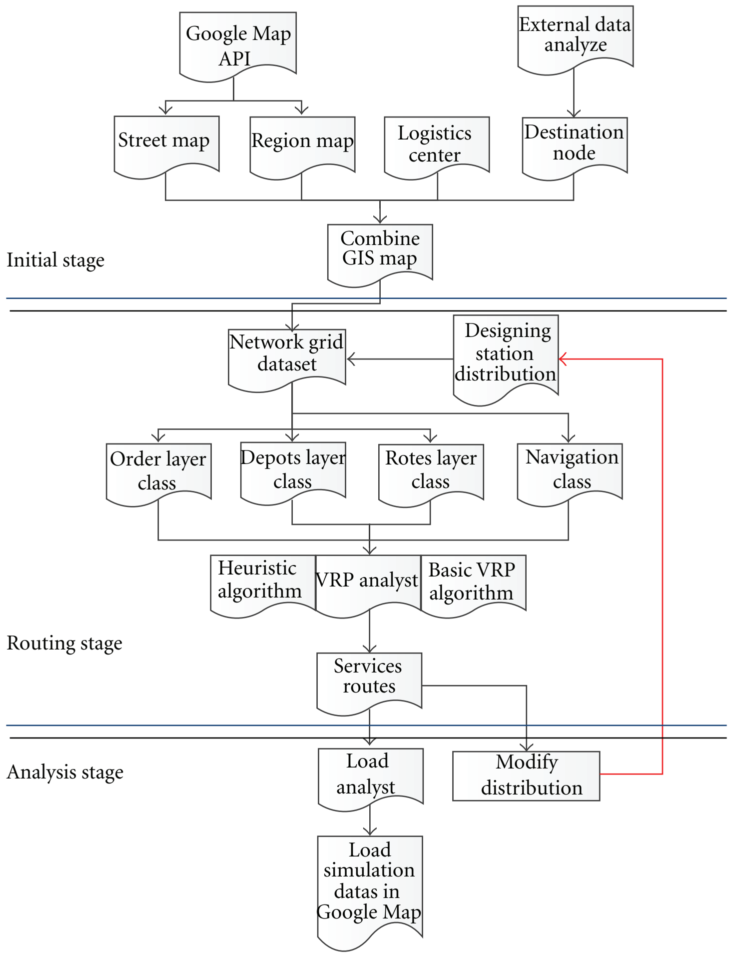



FIGURE 10: The simulation stage.

FIGURE 11: Information services for bicycle sharing system websites.

The rental information definition of PBS stations was listed below:

(1) the station ID, names,

(2) location information (latitude, longitude),

(3) the number of available bikes that cyclists can hire,

(4) the number of free docks that cyclists can return.

Our system collected the above information of Taipei, Kaohsiung, and Washington PBS stations from the Internet and provided options for users to choose.

Figure 12: The system interface.

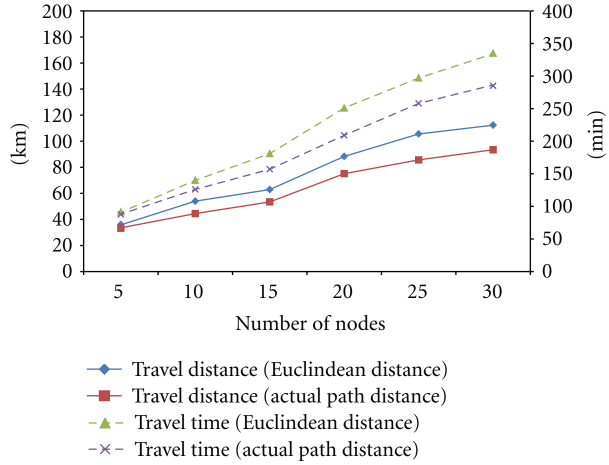



Figure 13: The nearest neighbor algorithm.

The system was implemented using the Microsoft. NET Framework in C# language. The system is a standard Windows application where the software feature configuration is simple and easy. The main display block was constituted by the map panel of Google Maps as shown in Figure 12. A sidebar displayed at the right-hand of the map panel offered seven main items: Map, Parameter, Simulation Engine, Info, Setting, and Cash. Each menu had different functions for simulation and intuitive operation via mouse-drop to modify the station location on the map area.

In addition to the above features, our PBS redistribution system included different menu items with several parameter data types:

(i) the simulation type: the TSP or the VRP,

(ii) the station state: a location and redistribution configuration value,

(iii) the vehicle: vehicle capacity, the maximum number of working hours per day,

(iv) the staff: the average time of redistributing one bike.

A PBS planner could adjust the simulation parameters by using appropriate items that were displayed with the right-hand sidebar. The planner can choose "Parameter" menu in the sidebar. Then, the planner must enter the data about redistributing vehicles and human resources. The simulation parameters will affect the final simulation results.

The redistributing routes of vehicles are constrained by the state of cycle hire docking stations, the full capacity of vehicles, and the working time. If the total transportation time of a vehicle is equal to or greater than the working time, the system will dispatch another vehicle to serve the remaining stations.

Our system estimates the number of vehicles and the redistributing routes that satisfy the constraints such as the capability limit of vehicles, prohibited turns, and one-way streets. The route then will be presented on Google Maps with the suggested order of visited stations.

Through the system, the planner could do the route planning efficiently and find the best scheduling strategy to achieve the hotspot analysis and the adjustment of the station deployment strategy to reduce the PBS operation cost.

## 5. Experimental Results

In this section, experiments were carried out on an X64 PC with Intel Core i5 2.6 GHz CPU. The performance of the algorithms for the TSP and the VRP was tested using three different construction methods with/without our actual path distance optimization method.

### 5.1. The Evaluation of Different Methods

*5.1. The Evaluation of Different Methods.* In Figures 13 to 15, we evaluated the travel distance and time for different numbers of nodes for 5 to 30 nodes with 10 km radius in the downtown of Taipei. We chose the nearest neighbor algorithm, the savings method, and the farthest insertion algorithm as construction algorithms and compared them with our actual path distance optimization method to solve the TSP.

In Figures 13 to 15, we can see that when the number of nodes increases, the travel distance and time increase. According to the results, when we used the actual path distance, we got shorter travel distance and time than when using the Euclidean distance. However, if the number of nodes had been too small (less than 15 nodes), the improvements of our method would have been less in some circumstances. In Figure 14, when the number of nodes is 25, the outcome of using the Euclidean distance is also the same as that of using our scheme. It happened so it found the optimum solution using the Euclidean distance. The location of the node distribution was another factor that affected the improvement due to road conditions, traffic regulations, and geographical factors. Based on Figures 13 to 15, we can know that the traffic in Taipei's downtown is not ideal. The average speed of vehicles (Travel Distance/Travel Time) is about 21 km/hour. The PBS planner should consider more about road conditions.

As shown in Figure 16, we compared construction methods with the actual path distance. We found that the farthest

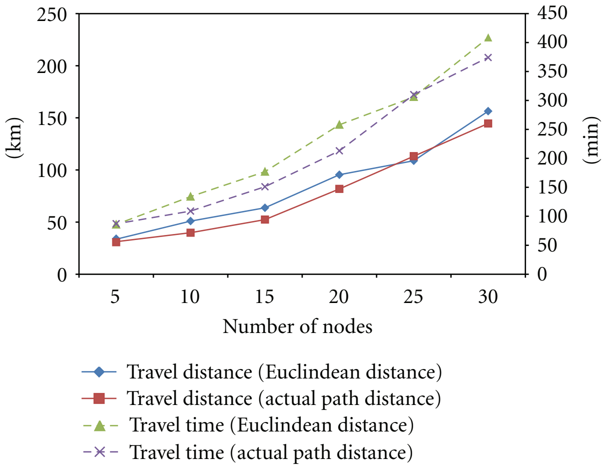



FIGURE 14: The savings method.

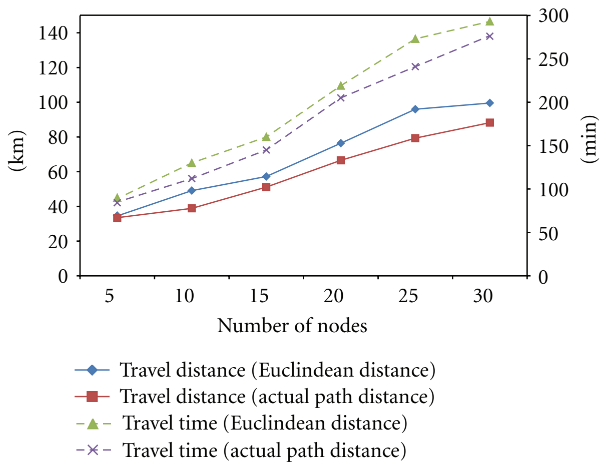



FIGURE 15: The farthest insertion algorithm.

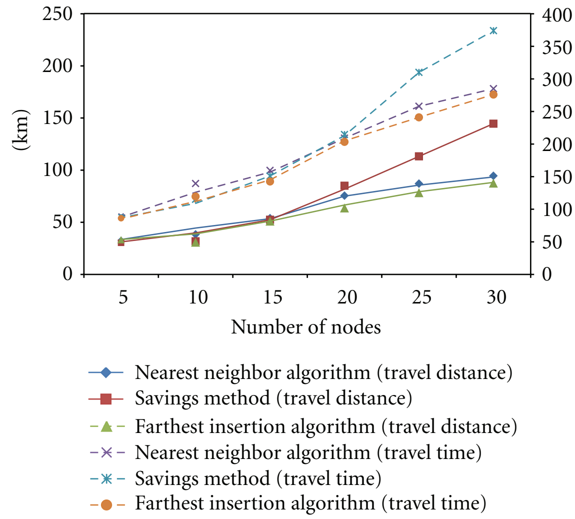



FIGURE 16: The comparison of construction methods.

insertion algorithm was the best solution in most cases. The algorithm considered the insertion position to constitute the minimum tour cost. The nearest neighbor algorithm became worse when the number of nodes increased. It could quickly find a short tour for the TSP, but the solution was not the optimal one. It only considered one node (the next nearest node) and ignored the effect of the other unvisited nodes that resulted in a longer tour.

*5.2. PBS Simulation.* The PBS planner may want to know the effect of vehicle trips and route changes if he/she adjusts the safety stock level for each station, vehicle capacity, and employee working time. Analysis of the results for different parameters is important for the management to know the labor force and vehicle trips. Our system has the ability to simulate the PBS to achieve the task.

In Table 2, we used the real data in Kaohsiung C-Bike [28]. We assumed that working time was 8 hours. The full capacity of vehicles was 40 bikes, while the staff prefetched 15 bikes and besides, it took 2 minutes to redistribute one bike. More detailed parameters were shown in Table 2.

We simulated the nearest neighbor algorithm with/without our scheme. In this simulation case, without our scheme, it caused more than 10% total travel distance and it easily exceeded the working time like truck No. 1 because the estimated time (the Euclidean distance/the average speed in a urban area) was unreliable. There may be additional operating expenses in working overtime.

Then, we made an interesting experiment. The planner wanted to replace full time employees with part-time employees. To simulate this case, we assumed that part-time employees' working time was two hours and other parameters were the same as those in Table 2. With such experiment, the planner could adjust the number of employees within a specific time.

According to the result in Table 3, the total travel distance and work time became longer; hence, it was not a good idea to replace full-time employees with part-time employees. Although we need two more trucks in our scheme than in the original scheme, the working time of each truck does not exceed the limit of the two hours. Without our scheme, trucks no. 1 to 7 exceed the working time due to inaccurate estimation and it will cause that the planner has to defray overtime pay for the part-time employees.

Table 4 was another experiment and we used real-time station information data from Capital Bikeshare. We set the working time for 6 hours, vehicle capacity to 30 bikes, and prefetching to 10 bikes. It took 3 minutes to redistribute one bike. In addition, we set the safety stock level from 20% to 80% at a bike station. That stations did not satisfy the condition would need to perform redistribution services to adjust the bike stock to 45%. According to the above definitions, there were only 41 stations needing to be redistributed. In this case, we used the nearest neighbor algorithm and the savings method to simulate redistribution services. Therefore, managers could benefit from these results to arrange vehicle trips and the route planning better. In our system, it was interesting to analyze the impact of changes in different parameters (e.g., working time, the capacity of

Table 2: Simulation results for the VRP of Kaohsiung C-Bike (1).

| The simulation case: Kaohsiung C-Bike (36/46 stations for service) | | | | |
|---|---|---|---|---|
| Parameters | Working time: 8 h<br>The full capacity of vehicles: 40 bikes<br>Prefetch: 15 bikes<br>The average time of redistributing one bike: 2 min<br>Station safety stock: upper limit (70%), lower limit (30%), adjust value (45%) | | | |
| | | Duration (h:min) | Route length (km) | Transit time (min) |
| The Nearest Neighbor Algorithm for the VRP (The Euclidean Distance) | Truck no. 1 | 8:18 | 48.450 | 190 |
| | Truck no. 2 | 7:39 | 86.475 | 297 |
| | Total distance: 134.925 km | | Total duration: 15:57 | |
| The Nearest Neighbor Algorithm for the VRP (The Actual Distance Path) | Truck no. 1 | 7:46 | 38.805 | 153 |
| | Truck no. 2 | 7:57 | 92.384 | 315 |
| | Total distance: 131.19 km | | Total duration: 15:43 | |

Table 3: Simulation results for the VRP of Kaohsiung C-Bike (2).

| The simulation case: Kaohsiung C-Bike (36/46 stations for service) | | | | |
|---|---|---|---|---|
| Parameters | Working time: 2 h<br>The full capacity of vehicles: 40 bikes<br>Prefetch: 15 bikes<br>The average time of redistributing one bike: 2 min<br>Station safety stock: upper limit (70%), lower limit (30%), adjust value (45%) | | | |
| | | Duration (h:min) | Route length (km) | Transit time (min) |
| The Nearest Neighbor Algorithm for the VRP (The Euclidean Distance) | Truck no. 1 | 2:01 | 11.404 | 43 |
| | Truck no. 2 | 2:14 | 14.309 | 60 |
| | Truck no. 3 | 2:18 | 17.863 | 74 |
| | Truck no. 4 | 2:10 | 15.674 | 54 |
| | Truck no. 5 | 2:15 | 21.984 | 83 |
| | Truck no. 6 | 2:10 | 21.488 | 80 |
| | Truck no. 7 | 2:15 | 26.952 | 93 |
| | Truck no. 8 | 1:49 | 26.852 | 80 |
| | Truck no. 9 | 0:56 | 15.526 | 44 |
| | Total distance: 172.052 km | | Total duration: 18:08 | |
| The Nearest Neighbor Algorithm for the VRP (The Actual Distance Path) | Truck no. 1 | 1:45 | 6.479 | 26 |
| | Truck no. 2 | 1:55 | 13.183 | 48 |
| | Truck no. 3 | 1:54 | 11.891 | 48 |
| | Truck no. 4 | 1:49 | 16.443 | 63 |
| | Truck no. 5 | 1:15 | 11.775 | 45 |
| | Truck no. 6 | 1:21 | 12.152 | 49 |
| | Truck no. 7 | 1:58 | 14.648 | 60 |
| | Truck no. 8 | 1:32 | 18.692 | 65 |
| | Truck no. 9 | 1:24 | 15.756 | 58 |
| | Truck no. 10 | 1:36 | 19.014 | 68 |
| | Truck no. 11 | 1:06 | 18.348 | 56 |
| | Total distance: 158.381 km | | Total duration: 17:35 | |

Table 4: Simulation results for the VRP of Washington Capital Bikeshare.

| The simulation case: Washington Capital Bikeshare (41/132 station for service) | | | | |
|---|---|---|---|---|
| Parameters | Working time: 6 h<br>The full capacity of vehicles: 30 bikes<br>Prefetch: 10 bikes<br>The average time of redistributing one bike: 3 min<br>Station safety stock: upper limit (80%), lower limit (20%), adjust value (50%) | | | |
| | | Duration (h:min) | Route length (km) | Transit time (min) |
| The Nearest Neighbor Algorithm for the VRP (The Euclidean Distance) | Truck no. 1 | 6:07 | 38.243 | 82 |
| | Truck no. 2 | 5:58 | 51.908 | 10 |
| | Truck no. 3 | 5:49 | 38.344 | 86 |
| | Truck no. 4 | 1:39 | 30.977 | 45 |
| | Total distance: 159.472 km | | Total duration: 19:33 | |
| The Nearest Neighbor Algorithm for the VRP (The Actual Distance Path) | Truck no. 1 | 5:58 | 34.650 | 83 |
| | Truck no. 2 | 5:54 | 36.470 | 78 |
| | Truck no. 3 | 5:55 | 45.628 | 94.6 |
| | Truck no. 4 | 1:40 | 31.340 | 47 |
| | Total distance: 148.088 km | | Total duration: 19:27 | |
| The Savings Method for the VRP (The Euclidean Distance) | Truck no. 1 | 7:33 | 95.938 | 195 |
| | Truck no. 2 | 7:03 | 72.764 | 147 |
| | Truck no. 3 | 7:18 | 69.341 | 144 |
| | Truck no. 4 | 0:43 | 10.601 | 14 |
| | Total distance: 248.644 km | | Total duration: 22:37 | |
| The Savings Method for the VRP (The Actual Distance Path) | Truck no. 1 | 5:31 | 61.545 | 121 |
| | Truck no. 2 | 5:37 | 45.338 | 95 |
| | Truck no. 3 | 5:18 | 46.072 | 102 |
| | Truck no. 4 | 3:55 | 31.463 | 68 |
| | Total distance: 184.418 km | | Total duration: 20:21 | |

vehicles, the safety stock level, adjust value). With these different scenarios, our system could help managers to find the best redistribution mechanism.

## 6. Conclusions

Bicycle redistribution is a part of the VRP. However, solutions to the VRP only use the Euclidean distance to calculate transport cost, which is unreliable. In this work, we proposed the actual path distance to replace the Euclidean distance. With real-world road information provided by Google Directions, transport cost can be calculated precisely to establish a better tour for bicycle redistribution.

Experimental results demonstrated that our scheme generates better vehicle routes, which have lower travel distance and time than the original schemes. Comparing different construction methods with our scheme, we found that the farthest insertion algorithm is the best in most cases and nearest neighbor algorithm becomes worse when the number of nodes increases. Hence, we recommend the PBS manager to use the farthest insertion algorithm with our scheme for bicycle redistribution.

We also implemented a system with GIS that integrates the Google Maps technology for renters and managers. The system uses a visualized interface to display the locations of bike rental sites and available bikes and bike parking docks for renters. The system also has the simulation ability to assist PBS managers to do the route planning efficiently and find the best scheduling strategy to achieve hotspot analysis, and it could adjust the strategy of the station deployment to reduce PBS operation cost.

## References

[1] A. Kaltenbrunner, R. Meza, J. Grivolla, J. Codina, and R. Banchs, “Urban cycles and mobility patterns: exploring and predicting trends in a bicycle-based public transport system,” *Pervasive and Mobile Computing*, vol. 6, no. 4, pp. 455–466, 2010.

[2] P. Midgley, “The role of smart bike-sharing systems in urban mobility,” *Journeys*, vol. 2, pp. 23–31, 2009.

[3] O. Bräysy and M. Gendreau, "Vehicle routing problem with time windows, Part I: route construction and local search algorithms," *Transportation Science*, vol. 39, no. 1, pp. 104–118, 2005.

[4] O. Bräysy and M. Gendreau, "Vehicle routing problem with time windows, Part II: metaheuristics," *Transportation Science*, vol. 39, no. 1, pp. 119–139, 2005.

[5] A. S. Bjarnadottir, *Solving the vehicle routing problem with genetic algorithms [M.S. thesis]*, Technical University of Denmark, Copenhagen, Denmark, 2004.

[6] Q. Z. Ye, "The singed euclidean distance transform and its application," in *Proceedings of the 9th International Conference on Pattern Recognition*, pp. 495–499, 1988.

[7] L. Rongliang and S. Yunru, "The design and implementation of public bike information system based on google maps," in *Proceedings of the International Conference on Environmental Science and Information Application Technology (ESIAT '09)*, pp. 156–159, July 2009.

[8] R. Luo, T. Wu, and Z. Wang, "WebGIS systematic research of public bicycles based on AJAX," in *Proceedings of the International Conference on Educational and Information Technology (ICEIT '10)*, pp. V138–V140, September 2010.

[9] Transport Canada, *Bike Sharing Guide*, Transport Canada, Ontario, Canada, 2009.

[10] G. Nachouki and M. Quafafou, "MashUp web data sources and services based on semantic queries," *Information Systems*, vol. 36, no. 2, pp. 151–173, 2011.

[11] J. Yu, B. Benatallah, F. Casati, and F. Daniel, "Understanding mashup development," *IEEE Internet Computing*, vol. 12, no. 5, pp. 44–52, 2008.

[12] R. Hoar, "A personalized web based public transit information system with user feedback," in *Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems (ITSC '10)*, pp. 1807–1812, September 2010.

[13] H. Kawano, M. Kokai, and W. Yue, "GIS-based solution of vehicle scheduling and routing problems in day-care Center," in *Proceedings of The 7th International Symposium on Operations Research and Its Application (ISORA' 08)*, pp. 336–343, November 2008.

[14] P. Matis, "Decision support system for solving the street routing problem," *Transport*, vol. 23, no. 3, pp. 230–235, 2008.

[15] L. Santos, J. Coutinho-Rodrigues, and C. H. Antunes, "A web spatial decision support system for vehicle routing using Google Maps," *Decision Support Systems*, vol. 51, no. 1, pp. 1–9, 2011.

[16] J. G. Su, M. Winters, M. Nunes, and M. Brauer, "Designing a route planner to facilitate and promote cycling in Metro Vancouver, Canada," *Transportation Research Part A*, vol. 44, no. 7, pp. 495–505, 2010.

[17] Y. J. Wu, Y. Wang, and D. Qian, "A Google-map-based arterial traffic information system," in *Proceedings of the 10th International IEEE Conference on Intelligent Transportation Systems (ITSC '07)*, pp. 968–973, October 2007.

[18] M. Jünger, G. Reinelt, and G. Rinaldi, "Chapter 4 The traveling salesman problem," *Handbooks in Operations Research and Management Science*, vol. 7, no. C, pp. 225–330, 1995.

[19] G. Laporte and F. Semet, "Classical heuristics for the vehicle routing problem," Tech. Rep. G-98-54, GERAD, Quebec, Canada, 1999.

[20] R. He, W. Xu, Y. Wang, and W. Zhan, "A route-nearest neighbor algorithm for large-scale vehicle routing problem," in *Proceedings of the 3rd International Symposium on Intelligent Information Technology and Security Informatics (IITSI '10)*, pp. 390–393, April 2010.

[21] G. W. Nurcahyo, R. A. Alias, S. M. Shamsuddin, and M. N. Sap, "Sweep algorithm in vehicle routing problem for public transport," *Jurnal Antarabangsa*, vol. 2, pp. 51–64, 2002.

[22] P. E. Danielsson, "Euclidean distance mapping," *Computer Graphics and Image Processing*, vol. 14, no. 3, pp. 227–248, 1980.

[23] P. Felzenszwalb and D. Huttenlocher, "Distance transforms of sampled functions," Tech. Rep. 2004-1963, CIS-Cornell University, Ithaca, NY, USA, 2004.

[24] G. Clarke and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," *Operations Research*, vol. 12, no. 4, pp. 568–581, 1964.

[25] G. K. Rand, "The life and times of the savings method for vehicle routing problems," *ORiON*, vol. 25, no. 2, pp. 125–145, 2009.

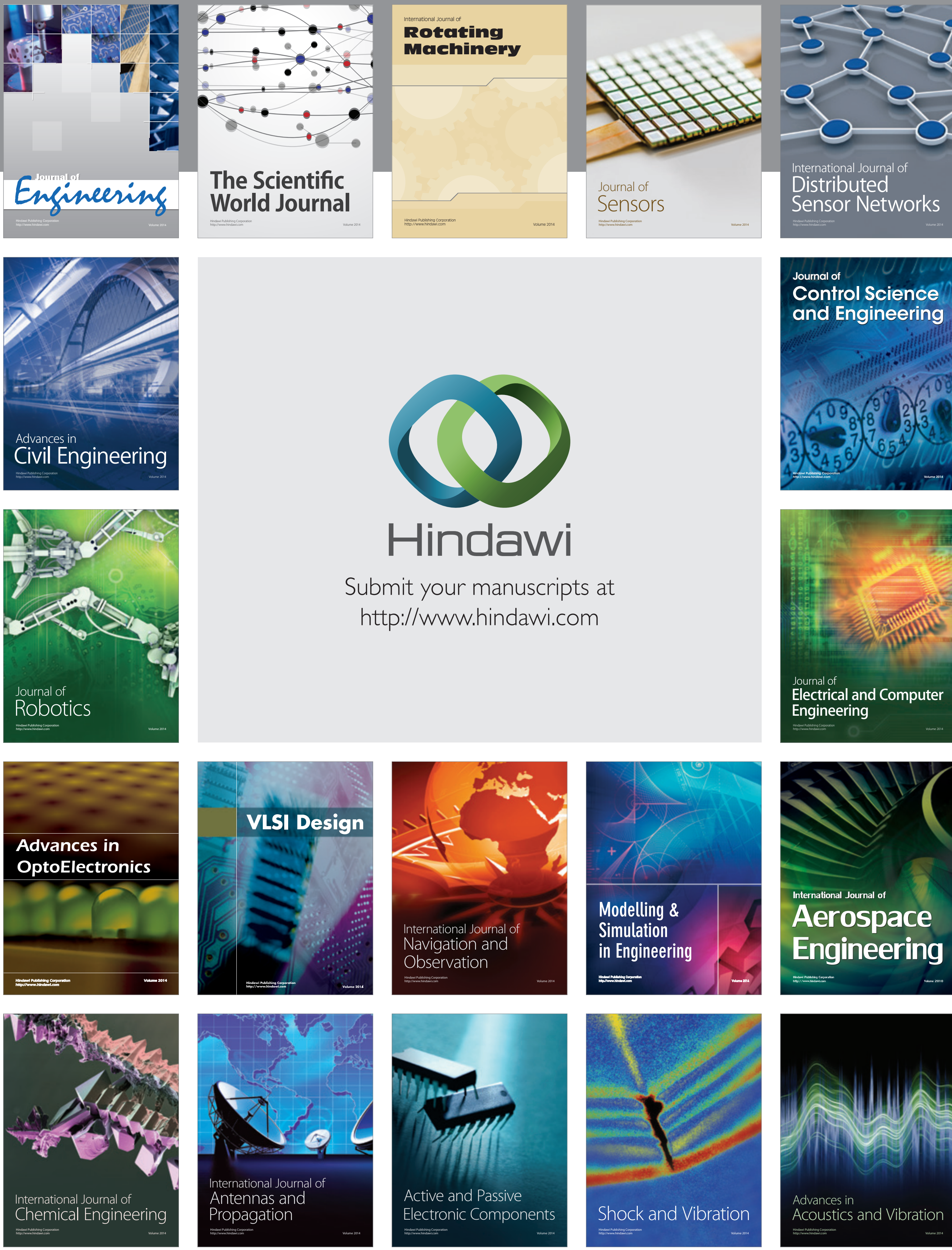
[26] D. J. Rosenkrantz, R. E. Stearns, and P. M. Lewis, "An analyis of several heuristics for the traveling Salesman problem," *SIAM Journal of Computing*, vol. 6, no. 3, pp. 563–581, 1977.

[27] W. Y. Chang, *A framework of automatic travel blog and re-navigation using mashup [M.S. thesis]*, Feng Chia University of Taiwan, Taichung, Taiwan, 2010.

[28] Kaohsiung City Public Bike, Taiwan, http://www.c-bike.com.tw/.

[29] Taipei YouBike, Taiwan, http://www.youbike.com.tw/.

[30] Capital Bikeshare, http://www.capitalbikeshare.com/.


Journal of Engineering
The Scientific World Journal
International Journal of Rotating Machinery
Journal of Sensors
International Journal of Distributed Sensor Networks
Advances in Civil Engineering
Hindawi
Submit your manuscripts at
http://www.hindawi.com
Journal of Control Science and Engineering
Journal of Robotics
Journal of Electrical and Computer Engineering
Advances in OptoElectronics
VLSI Design
International Journal of Navigation and Observation
Modelling & Simulation in Engineering
International Journal of Aerospace Engineering
International Journal of Chemical Engineering
International Journal of Antennas and Propagation
Active and Passive Electronic Components
Shock and Vibration
Advances in Acoustics and Vibration