

ACTIVITY-BASED COSTING FOR VEHICLE ROUTING PROBLEMS

A.J. Moolman^{1*}, K.Koen² and J. v.d. Westhuizen³

¹Department of Industrial Engineering
University of Pretoria, South Africa
alwyn@e-Logics.co.za

²Department of Business Management
University of Pretoria, South Africa
kkoen@clover.co.za

³e-Logics Pty Ltd
South Africa
johann@e-Logics.co.za

ABSTRACT

Activity-based costing (ABC) is a costing model that identifies activity costs in an organisation. It assigns the cost of activity resources to generate the actual cost of products in order to eliminate unprofitable products and to lower the prices of overpriced ones. The vehicle routing problem (VRP) is a combinatorial optimisation and nonlinear programming problem that seeks to service a number of customers with a fleet of vehicles in a cost-effective manner. In this article we propose a new approach to determine costing for vehicle routing type problems. The methodology incorporates the predictive sharing of a resource by clustering producers.

OPSOMMING

'Activity-based costing' (ABC) is 'n kostemodel wat die aktiwiteitskoste in 'n organisasie identifiseer. Dit allokeer die koste van die bronne sodat die ware koste van die vervaardiging en dienste van die produk bereken kan word om winsgewendheid te bepaal. Die 'vehicle routing problem' (VRP) is 'n kombinatoriese optimisering en nie-lineêre programmeringsprobleem wat verskeie kliënte met 'n vloot voertuie in die mees koste-effektiewe manier bedien. Die artikel bespreek 'n nuwe metode om die kombinasie van probleme op te los. Die metode maak gebruik van groeperingsalgoritmes om meer akkurate voertuig deling te voorspel.

¹ The author was enrolled for a PhD (Industrial Engineering) degree in the Department of Industrial Engineering, University of Pretoria.

²The author was enrolled for a Master Programme in Supply Chain Management (Business Management) certificate in the Department of Business Management, University of Pretoria.

*Corresponding author.

1. INTRODUCTION

One of the most significant problems of supply chain management is the distribution of products between locations. Known as the vehicle routing problem (VRP), this is one of the most challenging problems in the field of combinatorial optimisation. Dantzig and Ramser first introduced the VRP in 1959 (Dantzig and Ramser, 1959). It consisted of one source and many delivery stops. They proposed the first mathematical programming formulation for what then became known as the VRP. There has since been a steady evolution in the design of solution methodologies, both exact and approximate, for this problem. In 1964 Clarke and Wright proposed an effective greedy heuristic that improved the approach of Dantzig and Ramser. Since then, hundreds of models and algorithms have been proposed for the optimal and approximate solution of the different versions of the VRP. Solving the vehicle routing problem is a complex task, resulting in time-consuming algorithms.

In a business organisation, the activity-based costing methodology assigns an organisation's resource costs through activities to the products and services provided to its customers. It is generally used as a tool for understanding product and customer cost and profitability, and has mainly been used to support strategic decisions such as pricing, outsourcing, and identification and measurement of process improvement initiatives. The concepts of ABC were developed in the manufacturing sector of the United States during the 1970s and 1980s, when the Consortium for Advanced Manufacturing-International, now known simply as CAM-I, provided a formative role for studying and formalising the principles that have become more formally known as activity-based costing (CAM-I, n.d.).

We consider a problem of collecting products from producers and delivering them to a depot. From the depot, the product is transported to its final destination. We propose a solution to solve the ABC for problems that contain a VRP solution to determine allocation to resources. We conclude, using the ABC answer to determine an initial solution for the VRP. The outcome of the method is *activity-based costing for the raw material*.

Additional information gathered from the exercise includes:

- Allocation of producers to depots based on ABC;
- Possible initial solution for the VRP.

We will use the problem of supplying raw material from its source to the distribution centre, and then on to its final destination. We will not use the standard transportation problem solution, because of complexities added to the problem. We define some of the additional complexities as:

- Raw material transport to the depot shares a resource, i.e. a vehicle;
- We cannot determine the allocation of producers to the resource before the VRP has been solved, which is not part of the goal;
- Producers are not allocated to final destinations.

The previous points serve as confirmation that we need to make some assumptions about the solution in order to get to the goal.

2. LITERATURE

There is ample literature on ABC and on VRP. Applying the VRP influence on ABC is not generally used because of the complexity of the VRP. Using the ABC result as an initial solution for VRP seems to be a new approach in the VRP environment.

2.1. ABC

Activity-based costing was first clearly defined in 1987 by Robert S. Kaplan and W. Brunsina chapter in their book *Accounting and management: A field study perspective*. They initially

focused on the manufacturing industry, where increasing technology and productivity improvements had reduced the relative proportion of the direct costs of labour and materials, but had increased the relative proportion of indirect costs. For example, increased automation had reduced labour (a direct cost) but increased depreciation (an indirect cost) (Kaplan, 1987).

In 1999 Gary Cokins wrote an article aimed at certified public accountants who have difficulty embracing activity-based costing. In 'Learning to love ABC', Cokins explains that activity-based costing usually works best with a *minimum* amount of detail and *estimated* cost figures. He backs this up: "Typically, when accountants try to apply ABC, they strive for a level of exactness that is both difficult to attain and time-consuming -and that eventually becomes the project's kiss of death".

In 2000 Cokins wrote another article entitled 'Overcoming the obstacles to implementing activity-based costing'. Cokins noted that "activity-based costing projects often fail because project managers ignore the cardinal rule: It is better to be approximately correct than to be precisely inaccurate. When it comes to ABC, close enough is not only good enough; close enough is often the secret to success". Cokins also noted that the use of average cost rates, the use of overly-detailed information, and the failure to connect information to action can also hinder ABC projects. By understanding these concepts, Cokins felt that CPAs can enhance their roles as business partners and consultants.

Douglas T. Hicks has said: "Any small or midsize organization can develop an ABC system. It doesn't require a great commitment of time or financial resources. Nor does it require the implementation of special software integrated into the general ledger- although for larger organizations that may be a benefit. It requires only that management view its operations through 'the lens of ABC' and create a model that will enable it to measure costs in accordance with that view".

Gary Cokins, director of industry for a noted ABC software and services firm, tends to agree with Hicks. In his book *Activity-based cost management: Making it work*, he predicted that "within 10 to 20 years, everyone will have some sort of ABC. It's a matter of when, not if" (Cengage, 2002).

Almost ten years later, we are experiencing the need to determine where cost is coming from, to be able to control it. The VRP-type scenarios require a better way of estimating, while still keeping it simple to align with the approximation principles stated by Cokins.

2.2. VRP

A key issue in transportation is the cost-efficient management of a vehicle fleet providing pick-up service to a given set of customers with known demands. Products are collected from a customer, delivered to a depot, and then transported to their final destination.

Every vehicle route starts at a depot, collects the load from customers on the route, and returns to the depot. The route must satisfy all constraints relating to the problem: capacity of the vehicle, time windows, etc. This class of logistic problems is known as the vehicle routing problem (VRP) and its objective is usually to minimise the overall distance travelled by the vehicles while servicing all the customers. The interest in VRP problems comes from their practical relevance as well as from the considerable difficulty in solving them exactly. In the field of combinatorial optimisation, the VRP is regarded as one of the most challenging problems. It is indeed NP-hard, so that the task of finding the best set of vehicle tours by solving optimisation models is computationally prohibitive for real-world applications. As a result, different types of heuristic methodologies are usually applied.

A tremendous amount of work in the field of vehicle routing and scheduling problems has been published in the past four decades. It is summarised in books and surveys (see Laporte, 1992; Desrosiers et al., 1995; Fisher, 1995; Bramel and Simchi-Levi, 1997;

Crainic and Laporte, 1998; Golden et al., 2008). Some research efforts were oriented towards the development and analysis of approximate heuristic techniques capable of solving real-size VRP problems. Bowerman et al. (1994) classified the heuristic approaches to the VRP into five classes:

- 1) cluster-first/route-second
- 2) route-first/cluster-second
- 3) savings/insertion
- 4) improvement/exchange, and
- 5) simpler mathematical programming representations through relaxing some constraints.

From the two clustering procedures, the cluster-first/route-second looks more effective. This algorithm first groups the nodes into clusters, assigns each cluster to a different vehicle, and finally finds the vehicle tour by solving the corresponding travelling salesman problem (TSP).

3. PROBLEM FORMULATION

A company collects products from producers that are geographically dispersed. The cost of the collection function needs to be recovered from the producers, reflecting the actual cost based on a realistic model. The model should not be cumbersome, and should be integrated with the financial information systems.

The model should allocate cost in a way that provides incentives for producers that are located closer to the market or depot, or that are located in a concentration of other producers, or that are high volume producers. This will reduce the unit cost of collection.

The basic VRP is given by a set of identical vehicles, a depot, a set of customers to be visited, and a directed network connecting the depot and customers. Let us assume there are K vehicles, $V = \{0, 1, 2, 3, \dots, K-1\}$, and $N+1$ customers, $C = \{0, 1, 2, 3, \dots, N\}$. We denote the depot as customer 0, or C_0 . Each arc in the network corresponds to a connection between two nodes. A route is defined as starting from the depot, going through a number of customers, and ending at the depot. A cost c_{ij} and a travel time t_{ij} are associated with each arc of the network. The VRP has a finite number of feasible solutions. The solution space increases exponentially as the number of customers increases. Thus the VRP is known as a non-polynomial hard (NP-hard) problem.

The landed cost for this problem is defined as

$$\sum_{i=1}^N \sum_{j=1}^N \sum_{k=0}^{K-1} c_{ij} x_{ijk} + \sum_{b=0}^{B-1} c_{df} x_{dfb}$$

The first term represents the total cost to get the product to the depot. The second term represents the cost of the inter-branch vehicles between the depot and the factory. The decision variable x_{ijk} is 1 if vehicle k travels from stop i to j . The decision variable x_{dfb} is 1 if inter-branch vehicle b travels from the *depot* to the *factory*. The formula is for one depot only, and should be applied for all depots in the problem space. The VRP consists of minimising the first term, with all additional constraints defined for the problem.

Adopting Cokins' principle of using a *minimum* amount of detail, it is clear that solving the VRP to get ABC is overdoing it. The dynamics of day-to-day operational routing contribute to the inadequacy (or over-complication) of this method, because each producer is not necessarily part of each day's solution. The problem is to find an adequate *minimum* amount of detail that is a good estimation of the VRP.

The technology required for the tool to allocate collection cost must be able to cater for all the parameters, such as the cost drivers and associated activities, the location of producers

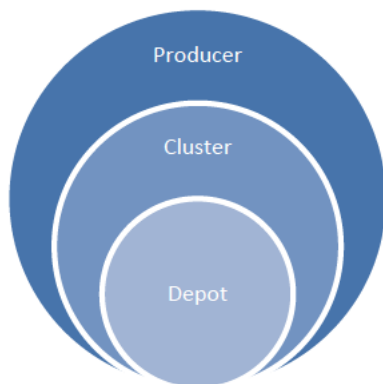
and distances between them, and the collection points and final destination. The cluster part of the 'cluster-first, route-second' VRP solution approach will be used as a starting point.

The clustering of producers into cost groups will be done via business rule-driven cluster algorithms, and the cost allocation should be modelled via scenarios where the cost factors can be compared and adjusted. This will result in the ability to create different scenarios for allocating collection costs to producers using density clustering algorithms, clustering settings, calculation weights, and performing cost allocation logic on the general ledger data.

3.1. Methodology

One of the heuristic approaches to the VRP is the 'cluster-first, route-second' method. Clustering is one of the major data-mining methods for knowledge discovery in large databases. It is the process of grouping large data sets according to their similarity.

A route resulting from a VRP solution can be seen as a cluster created through specific business rules. We reduce the constraints applied during a comprehensive VRP solution to provide a best-guess master route group. The cost allocation is done based on the following diagram.



Three factors influence the cost to a producer:

- 1) The cost from the producer to the centre of the cluster.
- 2) The cost from the centre of the cluster to the collection depot.
- 3) The cost from the producer to the collection depot.

The cost allocation algorithm can now exploit these values to provide a better *minimum* effort cost allocation.

3.2. Cost allocation factors

The calculation of landed cost is dependent on various cost factors. Cost at the depot can consist of:

- General ledger accounts
- General cost accounts
- Personnel cost

The cost per vehicle consists of:

- Fuel cost
- Fixed vehicle cost

- Variable vehicle cost

The solution is implemented as follows:

3.2.1. Scenario

The tool provides for multiple scenarios to be setup. Each scenario allows a change in:

- Producers
- Producers' volume
- Depots
- Factories (final destination)
- Vehicles
- Cluster parameters
- GL accounts
- Proportional allocation of GL accounts to producer cost factors.

3.2.2. Cluster

Clustering algorithms can be categorised into five main types, according to Han and Kamber (J. Han, 2001): *partitional*, *hierarchical*, *grid-based*, *model-based*, and *density-based* clustering algorithms.

In *partitional* algorithms, cluster similarity is measured in regard to the mean value of the objects in a cluster's centre of gravity (K-Means), or each cluster is represented by one of the objects of the cluster located near its centre (K-Medoid). K is an input parameter for these algorithms.

Hierarchical algorithms produce a set of nested clusters organised as a hierarchical tree. *Grid-based* algorithms are based on a multiple-level grid structure on which all operations for clustering are performed. In *model-based* algorithms, a model is hypothesised for each of the clusters, and the idea is to find the best fit of that model to others.

The *density-based* notion is a common approach to clustering. Density-based clustering algorithms are based on the idea that objects that form a dense region should be grouped together into one cluster. They use a fixed threshold value to determine dense regions. They search for regions of high density in a feature space that are separated by regions of lower density.

We have implemented a *partitional* and *density-based* cluster algorithm as an option for the user.

3.2.3. Partitional clustering

The main advantage of the partitional clustering in this solution is the option available for the user to decide the number of clusters. The solution implements the *K-means* and *K-medoid* methods.

K-means uses the centroid of the cluster, which normally represents a location, to determine the relation between the K clusters. The *K-means* algorithm relies on the ability to calculate a cost between the artificial cluster centroid and other points.

K-medoid is similar to the *K-means*, except that the centre point of the cluster is never artificial, but is represented by a point in the cluster. This allows for building clusters on cost functions which results can only be represented to a point in the data set- e.g. the distance that is read from a matrix or using the time between points.

The *K-medoid* algorithm can be described as follows:

- (1) The algorithm begins with the arbitrary selection of the k objects as medoid points out of n data points ($n > k$)
- (2) After selection of the k medoid points, associate each data object in the given data set to the most similar medoid. The similarity here is defined using a distance (cost) measure that can be Euclidean distance, Manhattan distance, or Minkowski distance. We interpret distance as a cost function.
- (3) Randomly select non-medoid object.
- (4) Compute total cost S of swapping initial medoid object to randomly selected non-medoid object.
- (5) If $S < 0$, then swap initial medoid with the new one. (If $S < 0$ then there will be new set of medoids.)
- (6) Repeat the steps until there is no change in the medoid.

3.2.4. Density-based clustering

The density-based clustering methods in the solution allow the user to identify *noise*. The key idea of the DBSCAN algorithm is that, for each point of a cluster, the neighbourhood of a given radius has to contain at least a minimum number of points - that is, the density in the neighbourhood has to exceed some predefined threshold.

The DBSCAN algorithm depends on the calculation of a centroid for a cluster. This is traditionally calculated as the average of x and y in a Euclidian space. The shared nearest neighbour (SNN) algorithm does not depend on a centroid for a cluster. It defines the similarity between points by looking at the number of nearest neighbours that two points share.

Using this similarity measure in the SNN algorithm, the density is defined as the sum of the similarities of the nearest neighbours of a point. Points with high density become core points, while points with low density represent noise points. All remaining points that are strongly similar to a specific core point will represent a new cluster.

The SNN algorithm can be described as follows:

- 1) Identify the k nearest neighbours for each point
- 2) Calculate SNN similarity between points
- 3) Calculate SNN density of each point
- 4) Detect core points
- 5) Form cluster from core points
- 6) Identify noise points
- 7) Assign the remaining points to the cluster that contains the most similar core point

3.2.5. Calculation

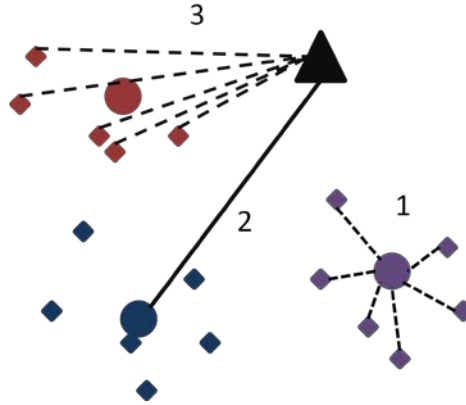
The cost factors mentioned earlier are now arranged as follows:

- Personnel cost - influenced by load time and travel time
- Vehicle variable cost - influenced by distance travelled
- Vehicle fixed cost - cost per vehicle
- Fuel cost - influenced by distance travelled

The landed cost per unit can now be calculated as a per cost factor. We assume that all time-bucket dependent values are given for the same unit: e.g., the personnel cost is for a day, the total distance is for a day, etc.

The problem arises because the producer shares a resource to deliver the product to the final destination. We estimate this sharing with the use of the constructed clusters.

Although this can never be as accurate as a solved VRP, the estimation implies better results than just a total weighted average.



The parameters used in the calculation relate to:

- 1) The distance from a producer to the centroid of the cluster. This assists in the relative position of the producer in the cluster. The influence should be low, but still valid, specifically for producers that are defined as noise, but have to be allocated to a cluster.
- 2) The distance from the cluster centroid to the depot can be the best estimation for cost shared by the producers.
- 3) Producer distance to depot is used to balance the fact that clustering is not an exact solution, and the individual producer's location still impacts landed cost.

Setting the influence of each distance determines the balance of the solution. The following section describes the formulas for some of the costs to the producer.

Personnel cost:

$$P_0 = \frac{T_0}{T_0 + T_f} P$$

where

P = total personnel cost

P_0 = depot allocated personnel cost

T_0 = total calculated depot times

T_f = total calculated inter-branch time

$$T_0 = \sum_{i=1}^N \sum_{k=0}^{R-1} t_{ik} x_{ik} + \sum_{k=0}^{R-1} t_{k0} + \sum_{i=1}^N f_i$$

The first term represents the travel time from a producer to the centroid of the cluster. $x_{ik} = 1$ if i belongs to cluster k . The second term represents the time from the cluster centroid to the depot. The third term represents the time spent at a producer, which can consist of a fixed time plus a variable time that depends on the volume.

The personnel cost part of the landed cost value per producer can now be calculated as follows:

$$PL_i = \frac{P_f}{m_k} + \left(\frac{t_{ik} + f_i}{T_k} P_k \right) / m_i$$

PL_i = landed cost per unit at producer i

P_f = personnel cost allocated to inter-branch

m_k = volume for inter-branch vehicle ki
 t_{ik} = travel time from i to centroid of cluster k
 f_i = service time at producer i
 T_k = total calculated cluster times
 P_k = personnel cost allocated to cluster

$$P_k = \frac{T_k}{T_0} P_0$$

m_i = volume at producer i

Vehicle variable cost:

$$V_k = \frac{D_k}{D_0} V_0$$

$$D_k = \alpha \sum_{i=1}^N d_{ik} x_{ik} + \beta d_{k0} + \gamma \sum_{i=1}^N \sum_{k=0}^{R-1} d_{i0} x_{ik}$$

$$D_0 = \alpha \sum_{i=1}^N \sum_{k=0}^{R-1} d_{ik} x_{ik} + \beta \sum_{k=0}^{R-1} d_{k0} + \gamma \sum_{i=1}^N d_{i0}$$

where

D_k = distance value for cluster
 D_0 = distance value for depot
 α = factor of the producer to cluster value
 β = factor of the cluster to depot value
 γ = factor of the producer to depot value

The landed cost based on vehicle variable cost can now be defined as:

$$VL_i = \frac{m_i}{m_k} V_f c_{qf} + \frac{d_{ik}}{\sum_{j=1}^N d_{jk} x_{jk}} V_k c_{qk}$$

The purpose of this calculation is to get an adjusted distance travel by incorporating the cluster. Any vehicle cost cannot be assigned directly to the distance from the producer to the factory. In the formula, it seems that the distance is proportionate, but it is actually the per kilometre cost factor that is shared between the producers.

The values of α, β and γ determine the weight of each type of distance on the cost. If

$$\gamma = 1, \alpha = 0, \beta = 0$$

The formula (excluding the inter-branch part) reduces to:

$$VL_i = \frac{d_{ik}}{\sum_{j=1}^N d_{jk} x_{jk}} \frac{\sum_{i=1}^N \sum_{k=0}^{R-1} d_{i0} x_{ik}}{\sum_{i=1}^N d_{i0}}$$

which indicates that the cluster portion is dependent only on the distance of its producers to the depot, and the final producer value is calculated depending on where in the cluster the producer is located. Typical values for α, β and γ are:

$$\alpha = 0.2, \beta = 0.6, \gamma = 0.2$$

This configuration places the influence of distance from the producer mainly on the distance of the cluster centroid to the depot. The other two factors help to penalise producers that are not close to the cluster centroid and producers that are further from the depot. In this way, producers in the same cluster will have different variable costs.

Fuel cost:

Fuel cost is calculated similarly to vehicle variable cost. The fuel cost is removed from the variable cost, because fuel cost can be read from the general ledger entry. If not, the fuel cost can be added to the vehicle variable cost.

$$FL_i = \frac{F_f}{m_k} + \frac{d_{ik}}{\sum_{j=1}^N d_{jk} x_{jk}} F_k$$

Fixed vehicle cost:

The fixed vehicle cost is calculated from the volume of the producer.

$$GL_i = \frac{m_i}{\sum_{j=1}^N m_j} \sum_{k=0}^{Q-1} c_{qk}$$

The total landed cost can now be defined as:

$$L_i = PL_i + VL_i + FL_i + GL_i + Price$$

Where 'Price' is the price paid per unit of the product.

4. CONCLUSION

The ABC calculated in this article presents a more realistic representation of what actually happens in the real world. Although the ABC should be kept simple, sharing a vehicle among multiple producers must be incorporated, as it influences the cost allocation per producer.

This ABC provides the end-user with better strategic information, and decisions can be made on marketing efforts, allocation of factories, assignment of producers to different depots, and more.

The principle established here can be extended through the alteration of decision-making formulas. The values calculated by the cluster approach provide a base for other scenarios in the calculation of activity-based costing.

5. REFERENCES

- [1] Bianchi, L., Birattari, M., Manfrin, M., Mastrolilli, M., Paquete, L., Rossi-Doria, O., et al. (2004, March). Research on the Vehicle Routing Problem with Stochastic Demand. *Technical Report IDSIA, IDSIA-07-04*.
- [2] CAM-I. (2008) *Overview of CAM-I*. Consortium for Advanced Manufacturing-International: <http://www.cam-i.org>
- [3] Cengage, G. (2002). *Activity-Based Costing*. Encyclopedia of Small Business: <http://www.enotes.com/small-business-encyclopedia/>
- [4] Choi, E., Tcha, D. (2005). A column generation approach to the heterogeneous fleet vehicle routing problem. *Computers & Operations Research*, 34.
- [5] Dantzig, G., Ramser, J. (1959). The Truck Dispatching Problem. *Management science*, 6(1).
- [6] Dondo, R., Cerda, J. (2006). A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. *European Journal of Operational Research*, 176.

- [7] Ertöz, L., Steinbach, M., Kumar, V. (2003). Finding Clusters of Different Sizes, Shapes, and Densities in Noisy, High Dimensional Data. *SIAM International Conference on Data Mining*.
- [8] Ester, M., Kriegel, H., Sander, J., Xu, X. (1996). A Density Based Algorithm for Discovering Clusters in Large Spatial Database with Noise. *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*. Portland, Oregon: Institute for Computer Science, University of Munich.
- [9] Estivill-Castro, V. (2002). Why so many clustering algorithms - A Position Paper. *ACM SIGKDD Explorations Newsletter*, 4(1, p65).
- [10] Golden, B., Raghavan, S., Wasil, E. (2008). *The Vehicle Routing Problem, Latest Advances and New Challenges*. Springer.
- [11] Han, J., Kamber, M. (2001). *Data Mining Concepts and Techniques*. San Francisco, CA: Morgan Kaufmann Publishers.
- [12] Kaplan, K., Robert, S., Bruns, W. (1987). *Accounting and Management: A Field Study Perspective*. Harvard Business School Press.
- [13] Laporte, G. (2009, August). Fifty Years of Vehicle Routing. *Les Cahiers du GERAD*(G-2009-43).
- [14] Moreira, A., Santos, M., Carneiro, S. (2008). *Density based clustering algorithms*.CiteSeerX:
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.123.6125&rep=rep1&type=pdf>
- [15] Park, H.-S., Lee, J.-S., Jun, C.-H. (2006). A K-means-like algorithm for K-medoids clustering and its performance. *Proceedings of ICCIE*.
- [16] Peterson, J. (2002). *Clustering Overview*.CiteSeerX:
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.136.4170&rep=rep1&type=pdf>
- [17] Toth, P., Vigo, D. (2001). *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics Philadelphia, PA, USA.
- [18] Winston, W. (1994). *Operations Research: Applications and Algorithms, Third Edition*. California.
- [19] Yue, S., Li, P., Guo, J., Zhou, S. (2004). Using Greedy algorithm: DBSCAN revisited II. *Journal of Zhejiang University-Science*, 5(11).

