TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol. 16, No. 2, November 2015, pp. 355 ~ 362
DOI: 10.11591/telkomnika.v16i2.8800

Combination of Ant Colony Optimization with Local Triangular Kernel Clustering for Vehicle Routing Problem with Time Windows

Aina Musdholifah*, Rahman Indra Kesuma
Department of Computer Science and Electronics, FMIPA UGM, Yogyakarta
*Corresponding author, e-mail: aina_m@ugm.ac.id, rahman.indra.k@mail.ugm.ac.id

Abstract

VRP is a common problem that occurred in logistics, including determination a route of products from the source to the destination. VRPTW is variation of VRP that use routing concepts in the serving process at the certain time interval. Recently, many methods are used to solve this optimization problem, for example ACO. LTKC-ACO was developed to improve the ACO solutions that apply LTKC to obtain a number of classes that are considered as the candidate list in ACO. Local Search is also used to avoid ACO getting stuck in the local optimum. In this study, two types of LTKC-ACO are developed that's related to time windows parameter usage in clustering. The experimental result of 56 Solomon's datasets showed that LTKC-ACO can improve the ACO solutions on 73,21% of datasets and can out performed then the other methods, especially on the datasets that have longer scheduling of service time.

Keywords*: routing, vehicle routing problem with time windows, ant colony optimization, local triangular kernel clustering*

1. Introduction

In recent decade, industrial activity that growing continuously with increasingly wide of working area, can't be separated from the logistics. Logistics is a process including all the commodity activities, production, and its displacement from the commodity producers to the customers. When the logistic can be well managed, the industry will get the advantages in the efficiency of funding usage and the optimization of production and distribution with the obtained resources. Important element of the logistics process is the management of transportation. Routing is a common problem that can be found in the transportation management fields. Routing is the process that's used to determine a route of products or packages from source to the desired destination [1].

One of the combinatorial problems which related to commodity logistics is Vehicle Routing Problem (VRP). Vehicle Routing Problem with Time Windows (VRPTW) is a variant of VRP that has a number of products or commodities shipping from source (main depot or warehouse) to several different destinations by using a number of vehicles and there are additional restrictions on the service of a customer must be started and finished at a predefined time windows [2]. VRPTW have been widely applied in logistics, but theoretically, these issues still have not found the best solutions, in which the wide scope of the problem becomes one of the reasons for the difficulties in obtaining the optimum solutions [3].

The development of methods still continues until now to obtain the optimum solution in resolving the combinatorial optimization issues that still evolving and growing significantly. One of the methods that can be used to resolve the issues is Metaheuristics. This method finds a solution by combination of local search procedures and higher strategies to provide wide guidance on the search process to find a global solution [4, 5].

Ant Colony Optimization (ACO) is an algorithm that is inspired by the behavior of ants in finding sources of food, that ants traveled begins from or ends to the nest with the pheromones to influence other ants on choosing a particular travel route [5]. ACO has been applied as a single method or hybrid to solve a similar problem in [2], [6-8]. This algorithm has advantages in finding a good solution, can be run continuously, and be able to adapt with the changes [9]. However, it still has disadvantages on hard complexity and has long execution time [10]. These

Received August 13, 2015; Revised October 18, 2015; Accepted October 28, 2015

situation usually caused by the difficulties on determination of time required for the concentration of the solution, that's because ACO often getting stuck in a local optimum.

Local Triangular Kernel Clustering (LTKC) method [11] is a density-based clustering algorithm that's used to face the limitations of the existing clustering algorithms. LTKC can handle the problems with high-dimensional data processing, contain large amounts of noise, and consist of cluster with arbitrary condition and densities. LTKC using combination of two nonparametric density estimation procedures to determines the density of given data points. Combination of *k*-nearest neighbor and kernel density estimation are used in LTKC to find a number of classes, which amount is not given as input parameters.

VRPTW which is discussed in this study use the benchmark instances in [12], which has also been discussed and resolved by some previous studies. In [13, 14] and [15], the better solution are obtained, but less than 20% and the worse solution in more than 30% that are compared to the best-known result. While in [2] and [16] obtain the better solution than the three studies above, but with a greater number of vehicles, i.e. under 2 vehicles.

The new hybrid algorithm approach, LTKC-ACO, is proposed to improve the ability of ACO and to get the better methods than the other methods in previous study. LTKC is used to support in the problem resolving by giving the candidate list to ACO. Furthermore, ACO will find a solution by applying the local search method, in keeping ACO out of getting stack in a local optimum and to find a global solution.

2. The Purposed Algorithm

ACO was introduced by Dorigo and Stützel (2004), has been applied to many combinatorial optimization problems, quadratic assignments and dynamic problems. ACO is one of the variant of metaheuristic techniques which is inspired by behavior of ants in finding sources of food, using the pheromones to influence other ant in choosing a particular travel route [5]. ACO is considered suitable to solve VRPTW by apply updating pheromone trail procedure that is used to strengthen the finding of solutions in the environment around the best solution [6].

LTKC was proposed by [11] is density-based clustering that determines the density of data points using two nonparametric density estimation procedures i.e. K-Nearest Neighbor (KNN) and Kernel Density Estimation (KDE). LTKC uses Bayesian decision rule in order to set the objects in respective clusters. LTKC only requires one parameter input, which is the number of nearest neighbors, *k* [11].

In this paper, LTKC-ACO is proposed to improve the solution that's obtained by ACO, where LTKC is used in the first level to provide the candidate list to ACO in order to support the problems solving. In the next level, ACO will find the solution by applying the local search methods in order to avoid the ACO getting stuck in a local optimum.

In resolving the problem, the grouping of data points is handled by using LTKC at the beginning of the process, with the calculation of the kernel nearest-neighbor density function to each data point. LTKC will allocate each data point to one cluster, that can give the maximum value of the density function [11]. This process will be continuously repeated if the cluster members structure is still changing or maximum iteration has not been reached. The clustering results of LTKC are used as the candidate list in ACO.

Generally, the ACO algorithm was applied in LTKC-ACO, is same as the algorithm that has been used and described in [17] and [18]. But there are adaptation in the parameters determination in each iteration, the calculation of visibility and waiting time, the application of local search, and the utilization of global pheromone updating.

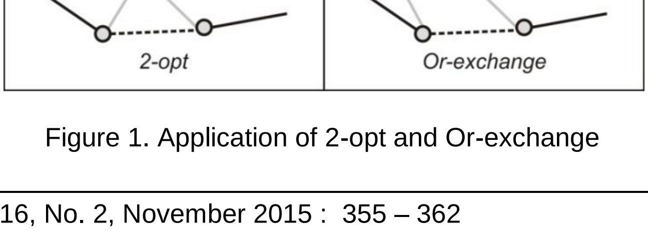

Figure 1. Application of 2-opt and Or-exchange

Local search improvement heuristics is an approach that can be generally used to searching or help finding the optimum solutions in hard optimization problems [19]. Local search is applied to explore around the initial solution environment, in order to improve the solution that can replaces the previous solution to be used in the next iteration [4]. In this study, the 2-optimality and or-exchange technique are used to improve the solution and as an effort to out of getting stuck in local optimum. 2-optimality is an iterative method that change the two nonconsecutive arcs and reconnection of four data points [15]. On the other side, or-exchange is the technique that inserts one, two or more successive arcs between two other data points in the same or different route [20].

Based on the illustration above, the LTKC-ACO algorithm that is used to solve VRPTW includes four main process are as follows:

1. Initialization
   a. Dataset which have the information about coordinate (x,y), open and closed service time, demand, and service time.
   b. Initialization the parameter i.e. *n* (number of data points), *k* (nearest neighbor), *maxIter* (number of ACO iteration), *ant* (number of ants), *ρ* (pheromones decay), *q_0* (exploitation versus exploration parameter), *g* (number of route that's taken to global pheromones updating), *α*, *β*, and *γ* (the importance of pheromones, visibility, and waiting time consecutively).
2. Normalization of attribute values using Formula (1)

$$x_i = \frac{x_i}{\max_{j\in N}(x_j)};\ y_i = \frac{y_i}{\max_{j\in N}(y_j)};\ a_i = \frac{a_i}{\max_{j\in N}(a_j)};\ b_i = \frac{b_i}{b_{n+1}} \qquad (1)$$

3. Clustering of data points using LTKC [11]
   a. Calculating the similarity of each data points using Formula (2), (3), (4) and put it into the similarity matrix.

$$dist_{coordinate}(A,B) = \sqrt[2]{(x_A - x_B)^2 + (y_A - y_B)^2} \qquad (2)$$

$$dist_{time\_windows}(A,B) = \sqrt[2]{(a_A - a_B)^2 + (b_A - b_B)^2} \qquad (3)$$

$$dist(A,B) = \frac{dist_{coordinate}(A,B)}{dist_{time\_windows}(A,B)} \qquad (4)$$

   b. Sorting the similarity matrix from the smallest to the largest (ascending).
   c. For each data points, take *k* nearest neighbor data points which is based on sorted similarity matrix and put into *listKNN.*
   d. For each data points (*x_i*),
      i. Calculating triangular kernel function of data point *x_i*, in the found *listKNN* that contain it.
      ii. Assigning data point *x_i* into the new cluster based on *listKNN* indices that gives the largest value of triangular kernel function.
   e. Resetting *listKNN* by taking *k* nearest neighbor data points from the new cluster.
   f. Checking the convergence condition, if the cluster structure was unable to change or the maximum iterations has been reached then LTKC process is stopped, otherwise back to step 3.d.
4. Finding the route by ACO
   a. For each ants, find a route by using the transition rule [17] that has an additional adaptation of the visibility calculation using Formula (5) and the waiting time calculation using Formula (6). The finding route process use the candidate list that is the result of LTKC.

$$\eta_{ij} = \frac{\left((maxDist\,.f) - \left(dist_{ij} + dist_{toZero}\right)\right)}{\left((maxDist.f) - (MinDist.f)\right)}; \qquad (5)$$

$$\text{where } f|dist_{j0} = \begin{cases} f=2 \mid dist_{toZero} = dist_{j0}, & \text{if } actual_{time} \geq 0{,}5\,.\,b_0 \\ f=1 \mid dist_{toZero} = 0, & otherwise \end{cases}$$

$$\omega_{ij} = \frac{maxWaitingTime - W_j}{maxWaitingTime};\ W_j = \begin{cases} a_j - r_j, & \text{if } (a_j - r_j) > 0 \\ 0, & \text{otherwise} \end{cases} \qquad (6)$$

b. After all the data points have been visited exactly once by an ant, apply local updates of pheromone trail [18].
c. After each ants get their route, apply local search (2-opt and or-exchange) to the five best route in one iteration.
d. Checking the best route globally and apply global updates of pheromone trail to the *g* best route in one iteration using Formula (7), (8) and (9).

$$\tau_{i,j}(t+1) = (1-\rho).\tau_{i,j} + \rho.\Delta\tau_{i,j}(t) \tag{7}$$

$$\Delta\tau_{i,j}(t) = \textstyle\sum_{k=1}^{g+1}\Delta\tau_{i,j}^{k} \tag{8}$$

$$\Delta\tau_{i,j}^{k}(t) = \begin{cases} \frac{1}{L_k(t)}, & \text{if } (i,j) \in T^k(t) \\ 0 \ , & \text{otherwise} \end{cases} \tag{9}$$

e. Checking the convergence condition, if the maximum iteration has not been reached then back to step 4.a.
f. Using the saved best route globally as the result of routing.

3. Research Methodology

3.1. Dataset

The experimental results of LTKC-ACO use classic set of 56 benchmark problems that are introduced in [12]. The datasets are divided into six categories and symbolized by R1, R2, C1, C2, RC1, and RC2. The characteristics of the used 56 datasets that can be seen in Table 1.

Tabel 1. The characteristics of VRPTW dataset in each category

Category	Number of Problems	Vehicle Capacity	Schedule Horizon	Service Time	Node Distribution	Time Windows
R1	12	230	230	10	Randomly	Short
R2	11	1000	1000	10	Randomly	Long
C1	9	200	1236	90	Clustered	Short
C2	8	700	3390	90	Clustered	Long
RC1	8	200	240	10	Combined	Short
RC2	8	1000	960	10	Combined	Long
Total	56					

All datasets contain the service of 100 customers with single main depot, and are informed about the vehicle capacity, the geographical position of the costumer, the demands, the time windows and the service time at each customer. Differences in the alphabetic code on dataset categories (R, C, and RC) describes the distribution related of data points (randomly distributed, have been clustered and a combination of both). While the numeric code on dataset categories (1 and 2) describe the scheduling horizon of customer service, where the code 1 has shorter scheduling horizon than the code 2.

3.2. Clustering Validation

In this study, internal clustering validation is used to evaluate the goodness of the LTKC results, which using the Silhouette Index measurement. The study in [21] explain that these measurement is started with finding the average distance of each data point M_i to the other data points in the same class (*a(i)*) using the formula (10). On the other side, the average distance of the data point M_i to the other data point in each different classes (*b(i)*) was evaluated using Formula (11).

$$a(i) = \frac{1}{n_k - 1}\sum_{\substack{i' \in I_k \\ i' \neq i}} d(M_i, M_{i'}) \tag{10}$$

$$b(i) = \min_{k' \neq k}\left(\frac{1}{n_{k'}}\sum_{i' \in I_{k'}} d(M_i, M_{i'})\right) \tag{11}$$

Where n_k is the number of cluster members of C_k, I_k is the indices of data that's contained in the class C_k and $C_{k'}$ is a class which does not contain M_i.

Furthermore, each data point M_i is calculated the silhouette width (*s(i)*) using the formula (12). The range of *s(i)* values is [-1, 1], where a value close to 1 indicates that the data point M_i has been assign in the right cluster.

$$s(i) = \frac{b(i)-a(i)}{\max(a(i),b(i))} \tag{12}$$

At the end, finding the average of silhouette width in one class (S_k) using Formula (13) and finding silhouette index global (C), which is the average of S_k, using Formula (14).

$$S_k = \frac{1}{n_k}\sum_{i\in I_k} s(i) \tag{13}$$

$$C = \frac{1}{K}\sum_{k=1}^{K} S_k \tag{14}$$

The optimum value of silhouette index global is that has the largest value of C.

3.3. Evaluation of Routing Results

The routing results of used dataset, are compared with ACO algorithms, the exact method [22], Tabu Search Heuristic [14], Multi-Objective Genetic Algorithms [16], Multiple Ant Colony System [6], and Hybrid Ant Colony System with Simulated Annealing [2]. The considered aspects in the comparison are the traveled distance of the route and the number of vehicles.

While the analysis process of LTKC-ACO to datasets were performed by observing the standard deviation of the results and the execution time. For understanding the influence of the time windows parameter usage in the clustering, LTKC-ACO is compared with the same algorithm, but without using the time windows parameter in the clustering.

4. Result and Analysis

LTKC-ACO requires seven parameters to obtain the optimum results, i.e. *k* nearest neighbors (for LTKC), the importance of pheromones (*α*), visibility (*β*), and the waiting time (*γ*) in the transition rule of ACO, the maximum ACO iteration (*maxIter*), the number of ants (*ant*), and the pheromones decay (*ρ*). The finding of optimum values to *k* parameter is performed in the range of [1, 80] from 100 used data points. In this study, was applied the two different types of LTKC i.e. LTKC that use a time windows parameter (LTKC-ACO Type 1) and that do not use time windows parameters (LTKC-ACO type 2).

Tabel 2. List of the best *k* value for LTKC without time windows

Datasets	*k* Value	Silhouette Index Value
R1-01	54	0.5242
RC-01	47	0.4567
C1-01	39	0.4781
C2-01	12	0.4996

The range of *k* parameter value for LTKC-ACO type 1 is [2, 9]. While the used value of *k* parameter for LTKC-ACO type 2 can be seen in Table 2. The parameter values-finding from the importance of pheromones, visibility and the waiting time for ACO transition rule, was performed by comparing their sequence that will more influence the transition rule. The optimum values which are obtained to those parameters are *α* = 0.5, *β* = 0.75, and *γ* = 1.0.

Furthermore, the finding of optimum values from *maxIter*, *ant* and *ρ* is performed by comparing the combination of those parameters with four different values. The *ant* parameter use the values of 5, 10, 15 and 20; *ρ* use the value of 0.1, 0.4, 0.6, and 0.9; while for *maxIter* should be adapted to the used dataset, but the iterations are still in the range of [1, 1000]. Table

3 show the list of best parameter combination from *maxIter*, *ant* and *ρ* in each category of the used dataset.

Tabel 3. List of best parameter combinations from *maxIter*, *ant* and *ρ*

Datasets	*maxIter*	*ant*	*ρ*
R1-01	750	20	0.1
R2-01	75	15	0.1
C1-01	100	10	0.9
C2-01	10	5	0.1
RC1-01	750	20	0.1
RC2-01	100	20	0.1

The experiment for one dataset is repeated for five times and implemented on a computer with Intel® Core™ i3 2.4 GHz processor and 2048 MB of RAM. The execution time is used as the boundary of the iterations number for LTKC-ACO and ACO, in order to get the fair comparison. From the experimental result, was observed that the average result of LTKC-ACO Type 2 is better than the ACO for all dataset categories, with the difference in the average number of vehicles are less than one. While on LTKC-ACO Type 1, there are three categories that have a worse of average result (C1, C2 and RC1) and the others get a better of average result. Then for the difference in the average number of vehicles which is used on LTKC-ACO type 1, there is one category (RC1) which has more than one (worse) and the others have less than one vehicle. These comparisons can be seen in Figure 2 and 3.

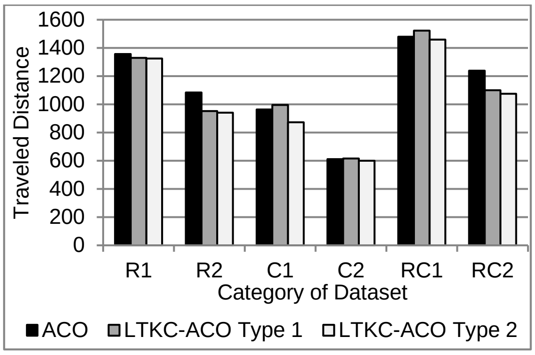

Figure 2. Comparison of LTKC-ACO with ACO on traveled distance

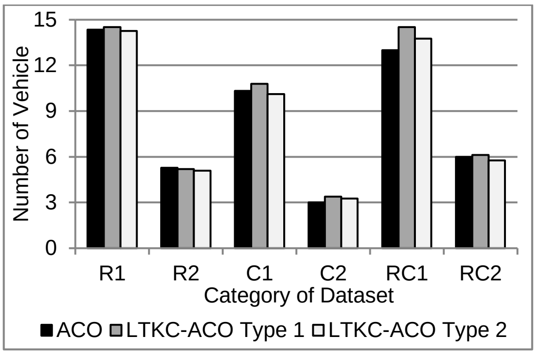

Figure 3. Comparison of LTKC-ACO with ACO on the number of vehicles

Furthermore, the comparison of the obtained results from LTKC-ACO Type 2, which is considered better than Type 1, with the exact method [22] of the 56 datasets can be seen in Figure 4. LTKC-ACO can give a good performed only in the dataset with type/code 2, which can obtain a better result at 15 out of 27 datasets with code 2. Likewise on the comparison with some methods in previous studies, LTKC-ACO is also can out performed only to the datasets with code 2. This is showed by the capability of LTKC-ACO to obtain the better average result, are based on a traveled distance aspects, from 2 to 3 algorithm on datasets with R2 and RC2 category, as shown in Table 4. In the R1 and RC1 category, LTKC-ACO still can't compete with some methods by obtain the worse solutions than them in the both aspects of travelled distance and number of vehicle. While LTKC-ACO was able to equal or close to the good ability of several methods in the categories of C1 and C2, with produce the difference of obtained result in the traveled distance less than 50 and the number of vehicles that not more than one vehicle.

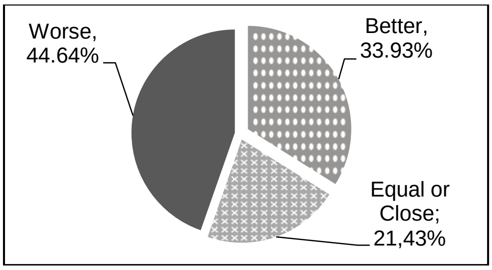

Figure 4. Percentage comparison of LTKC-ACO with exact methods

Tabel 4. Comparison of LTKC-ACO with methods from previous studies

Category of Dataset	LTKC-ACO	TS	MO-GA	MACS	ACO-SA
R1	1324,38 [a]	1210,14	1204,48	1217,73	1203,56
	14,25 [b]	12,08	13,1	12,00	12,83
	292,92 [c]	1200		1800	425
R2	939,52	969,57	893,03	967,75	932,23
	5,09	2,73	4,5 [3]	2,73	3,09
	316,09 [1]	3600 [2]		1800 [4]	437 [5]
C1	871,01	828,38	828,48	828,38	828,76
	10,11	10,00	10,00	10,00	10,00
	156,78	1200		1800	239
C2	599,70	589,86	590,60	589,86	589,86
	3,25	3,00	3,00	3,00	3,00
	96,50	3600		1800	363
RC1	1458,22	1389,78	1370,79	1382,42	1363,84
	13,75	11,50	13,00	11,63	12,50
	314,63	1200		1800	403
RC2	1074,86	1134,52	1025,31	1129,19	1079,81
	5,75	3,25	5,6	3,25	3,75
	234,88	3600		1800	370

a : traveled distance
b : number of vehicle
c : execution time (second)

1 : Intel® Core^{TM} i3 2,4 GHz 2048 MB Memory
2 : Sun Ultra 2 300 MHz
3 : Intel Pentium IV 1600 MHz 512 MB Memory
4 : Sun UltraSparc 1 167 MHz 70 Mflop/s
5 : Intel Pentium III 1000 MHz 128 MB Memory

The comparison of execution time can't be done in a fair, because the computers specification which is used for each method is different. This is appended with MO-GA algorithm which do not provides the information that's related to the execution time were used, but only provides the information about the number of generations that is 350 generations. So in this study, the execution time features only can be displayed to be a consideration for the next research, if needed.

5. Conclusion

LTKC-ACO Type 2, that do not use the time windows parameter in clustering, have a better capability than LTKC-ACO Type 1. This is showed by the performance of LTKC-ACO Type 2 to improve the quality of the ACO results on 41 datasets, 7 datasets is close or same, and 8 datasets is lower than ACO. However, when are compared with the other methods in previous studies, LTKC-ACO Type 2 only can out performed on the dataset with the code 2, that have longer scheduling of service time.

Future work will be involved by trying other clustering methods in order to provide a better *candidate list* for the ACO, representing VRPTW into the problems with multi-objective function, using *direct communication* (*antennation*) to enhance the capability of ACO, and using the other methods of the local search application to replace the 2-optimality, that are not able to provide significant improvements of the results when face the problems with short scheduling of service time.

References

[1] Babu BR, Poojary M, Renuka B. Application of Hybrid Ant Colony Optimization (HACO) Algorithm for Solving Capacitated Vehicle Routing Problem (CVRP). *International Journal of Computer Science and Information Technologies*. 2012; 3(2): 3540-3543.

[2] Chen CH, Ting CJ. A Hybrid Ant Colony System for Vehicle Routing Problem with Time Windows. *Journal of the Eastern Asia Society for Transportation Studies*. 2005; 6(206): 2822-2836.

[3] Fei W. Study on VRPTW Based on Improved Particle Swarm Optimization. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2014; 12(6): 4543-4549.

[4] Vidal T, Crainic TG, Gendreau M, Prins C. Heuristics for Multi-Attribute Vehicle Routing Problems: A Survey and Synthesis. Québec: CIRRELT. 2012.

[5] Santosa B, Willy P. Metoda Metaheuristik: Konsep dan Implementasi. Surabaya: CV Guna Widya. 2011.

[6] Gambardella LM, Traillard E, Agazzi G. MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows. In: Corne ID, Dorigo M, Glover F. *Editors*. New Ideas in Optimization. 1st ed. London: McGraw-Hill; 1999: 63-76.

[7] Reimann M. *Guiding ACO by Problem Relaxation: A Case Study on the Symmetric TSP*. 4th International Workshop on Hybrid Metaheuristics. Dortmund. 2007: 45-55.

[8] López-Ibáñez M, Blum C, Thiruvady D. *Beam-ACO Based on Stochastic Sampling for Makespan Optimization Concerning the TSP with Time Windows*. 9th European Conference – Evolutionary Computation in Combinatorial Optimization. Tübingen. 2009: 97-108.

[9] Jaiswal U, Aggarwal S. Ant Colony Optimization. *International Journal of Scientific and Engineering Research*. 2011; 2(7): 27-33.

[10] Setiawan A, Andriyanto F, Putro LS, Prakisya NPT, Perdana U. Perbandingan Algoritma Ant Colony Optimization, Disjktra, Tabu Search, Multiple Ant Colony System untuk Vehicle Routing Problem dengan Time Window. SKom Thesis. Surakarta: Jurusan Informatika FMIPA UNS; 2012.

[11] Musdholifah A, Mohd Hashim SZ. Cluster Analysis on High-Dimensional Data : A Comparison of Density-based Clustering Algorithms. *Australian Journal of Basic and Applied Sciences*. 2013; 7(2): 380-389.

[12] Solomon MM. Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. *Operations Research*. 1987; 35(2): 254-265.

[13] Shaw P. Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems. In: Mather M, Puget JF. *Editors*. In Principles and Practice of Constraint Programming. 1st ed. New York: Springer Science and Business Media, Inc.; 1998: 417-431.

[14] Cordeau JF, Laporte G, Mercier A. A Unified Tabu Search Heuristic for Vehicle Routing Problems with Time Windows. *Journal of the Operational Research Society*. 2001; 52(8): 928-936.

[15] Akeb H, Bouchakhchoukha A, Hifi M. *A Beam Search Based Algorithm for the Capacitated Vehicle Routing Problem with Time Windows*. Proceedings of the 2013 Federated Conference on Computer Science and Information Systems. Krakow. 2013: 329-336.

[16] Ombuki B, Ross BJ, Hanshar F. Multi-Objective Genetic Algorithms for Vehicle Routing Problem with Time Windows. *The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*. 2006; 24(1): 17-30.

[17] El Hassani AH, Bouhafs L, Koukam A. A Hybrid Ant Colony System Approach for the Capacitated Vehicle Routing Problem and the Capacitated Vehicle Routing Problem with Time Windows. In: Caric T, Gold H. *Editors*. Vehicle Routing Problem. 1st ed. Vienna: In-Teh; 2008: 57-70.

[18] Bonabeau E, Dorigo M, Theraulaz G. SWARM INTELLIGENCE - From Natural to Artificial Systems. New York: Oxford University Press. 1991.

[19] Qi C. Vehicle Routing Optimization in Logistics Distribution using Hybrid Ant Colony Algorithm. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11(9): 5308-5315.

[20] Savelsbergh AWP. An Efficient Implementation of Local Search Algorithms for Constrained Routing Problems. *European Journal of Operational Research*. 1990; 47(1): 75-85.

[21] Desgraupes B. Clustering Indices. Paris: Lab Modal'X - University Paris Ouest. 2013.

[22] Kallehauge B, Larsen J, Madsen OBG, Solomon MM. Vehicle Routing Problem with Time Windows. In: Desaulniers G, Desrosiers J, Solomon MM. *Editors*. Column Generation. 1st ed. New York: Springer Science and Business Media, Inc; 2005: 67-98.