■ 1208

Generating Travel Itinerary Using Ant Collony Optimization

Z.K. Abdurahman Baizal*, Aniq A Rahmawati, Kemas M Lhaksmana, Moh Z Mubarok, M. Qadrian
Department of Computational Science, Faculty of Informatics, Telkom University
Bandung, Indonesia, telp/fax +62-22-7564108
*Corresponding author, e-mail: baizal@telkomuniversity.ac.id[1], aniqatiqi@telkomuniversity.ac.id[2],
kemasmuslim@telkomuniversity.ac.id[3], muhammad.zidni.mubarok@gmail.com[4],
pumphead16@gmail.com[5]

Abstract
Travelling is one of the activities needed by everyone to overcome weariness. The number of information about the tourism destination on the internet sometimes does not provide easiness for oncoming tourists. This paper proposes a system capable of making travel itinerary, for tourists who want to visit an area within a few days. For generating itinerary, the system considers several criterias (Multi-criteria-based), which include the popularity level of tourist attractions to visit, tourist visits that minimize budgets or tourist visits with as many destinations as possible. To handle multi criteria-based itinerary, we use the concept of multi attribute utility theory (MAUT). The running time of multi criteria-based itinerary is not significantly different from time-based itinerary. In addition, the number of tourist attractions in the itinerary is more than time-based itinerary, because the combination of solutions from each ant becomes more diverse.

Keywords: travel itinerary, ant colony, E-tourism, travelling salesman problem, travel scheduling, travel recommender system

1. Introduction

Tourism is an important sector in Indonesia because it is able to introduce Indonesia in the international arena and generate a lot of foreign exchange. Indonesia has many tourism potentials, and Bali is the first tourist destination in Indonesia. Vihikan et al [1] has developed a method for predicting the number of tourists coming to Bali by using the recurrent neural network, to assist the government in determining the tourism strategy. Besides Bali, Bandung is also one of the leading tourist destination in Indonesia, so we take Bandung as the domain of the our system. In recent years, more and more tourists decide that Bandung city as a tourist attraction because Bandung has interesting and varied sites than other cities around. Not only offers the beauty of nature, the potential of local wisdom is able to attract the attention of tourists. Bandung tourism development also continues to be done not only physical improvements, but also the ease of information systems that can facilitate the tourists to look for everythings related to tourist destination so that information on the internet about tourist attractions in Bandung is increasingly found and continues to grow.

Lee et al [2], has created a system that can recommend tourist attractions in Taiwan and make a route from recommended tourist attractions, T. Q. Le and D. Pishva [3], make a recommendation of a tourist route in Tokyo, Osaka, Kyoto based on weather prediction, ongoing events and tourist's budget. Ardissono, Liliana, et al [4], make a recommendation of tourist attractions based on user's age. Pieter Vansteenwegen et al [5], made a route recommendation with regard to Point of Interest such as museums, churches, parks as reference in recommending tourist routes at Belgium for a few days. Meanwhile, B Rodríguez et al [6] uses Multi-Objective Metaheuristic using an Adaptive Memory Procedure (MOAMP) as the basis for selecting a tourist attraction. However, from the researchs mentioned above, there has no development of the system of tourism itinerary generation by considering some criteria of user need. Furthermore, this paper proposes a model for generating a travel itinerary, for a few days visit. The criterias (multi criteria) are: 1). Number of tourist attraction as much as possible, 2).

Popular tourist attractions (high rating), 3). Itinerary that minimizes budget. Currently, many models have been developed to build the travel itinerary by referring to the problem of finding the solution of the traveling salesman problem (TSP). The ant colony algorithm proved to be reliable in solving combinatorial problems [7-8], especially TSP [9]. Liu, Xiao-jun et al [10], also uses ant-colony for planning-process optimization of a machine so that less cost is needed. We use ant colony algorithm for generating this itinerary, by considering some constraints, such as open and closing hours of each tourist sites and the average of visiting time of travelers at each destination. The three criterias of user need are considered as user preferences, and the itinerary is generated according to these three criterias by utilizing multi attribute utility theory (MAUT) concept.

2. The Multi Criteria Case for Generating Itinerary

In our proposed model, itinerary is generated based on user preferences, in the form of three criteria as discussed in section 1. We use the Multi-Attribute Utility Theory (MAUT) concept to accommodate this multi-criteria case. According to Schaafer and Ralph [11], MAUT is a way of finding the value of a product by considering each attribute of the product and the priority scale of the attribute. Bichler et al [12] use MAUT to predict the winner in an auction. Meanwhile, Torrance et al [13] use MAUT to measure one's health status based on his daily life. Suppose an object will be evaluated using MAUT, so the evaluation can be formulated by,

$$u(x) = \sum_{i=1}^{n} w_i v_i(x) \tag{1}$$

Value $u(x)$ is an evaluation (utility value) of a tourist object, w_i is a coefficient used as a priority measure or Degree of Interest (DOI) of an attribute i, $\sum w_i = 1$, whereas $v_i(x)$ is the actual value of the attribute i of a object (For example: rating value or travel time). For example: the rating of x object is 0.8 (normalized), budget is 1.0 (normalized) and its travel time is 0.2 (normalized), with DOI of the rating is 0.6, the DOI of budget is 0.2 and the DOI of travel time is 0.2. Hence, then the utility value of x:

$$u(x) = doi_{rating} rating + doi_{distance} distance + doi_{budget} budget \tag{2}$$

In calculating utility values, it is necessary to normalize the value of each atribute, in order to equalize the value range of an attribute, as formulated below,

$$X_{norm} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{3}$$

The notation X represents the value of an attribute, $X_{\min}$ is the minimum value of attribute X in the data, $X_{\max}$ is the maximum value of attribute X in the data

3. The Ant Optimization for Solving Travelling Salesman Problem

In many works, travel itinerary determination is considered as a problem to find solutions for *travelling salesman problem* (TSP). We use ant colony algorithm to overcome this problem. This algorithm is introduced by Marco Dorigo [14], inspired by the behavior of ant colonies while feeding. Ants will move randomly to find food, when the ants return they will give a trace by releasing a pheromone substance, the others will recognize the trace and follow the path with the strongest pheromone.

```
Algorithm1.  AntCollonyForTSP(iteration,ants)
input: Total of iteration(iteration),  array of ants(ants)
1: i ← 0; n ← 0
2: while( i < iteration):
3:   while(n < length(ants)):
4:     ants[n].current ← random(1..ants[n].allowed)
5:     ants[n].tabu[] ← ant[n].current
6:     ants[n].allowed.remove(ants[n].current)
7:     for ( j ≤ total_node-1):
8:        ants[j].current ← next_node()
9:        ants[j].tabu[]←ant.current
10:       ants[j].allowed.remove(ants[n].current)
10:    update_pheromone()
11:     n++
12:   endwhile
13:   i++
14: endwhile
```

Algorithm 1 is the flow for the whole simulation for TSP completion by using ant-collony optimization [13] as follows,

3.1 Initialization process

A number of ants are initialized to be assigned randomly to each node that has been created, ant colonies are initialized because is is used to find the best solution space. In the ant variable, there are *maxcost*, *curentnode*, *tabu* (array to hold visited nodes) and *allowed* (array to hold unvisited nodes). In addition, the cost table is defined as the weight of the edge between the nodes. Next is pheromone initialization. This pheromone is used so that ants have a tendency to choose city *j* after city *i*. In other words, with this pheromone, ants may prefer the path passed by ants before. Initialization of pheromone generally uses the equation:

$$Pheromone_{ij} = \frac{1}{TotalNode} \tag{4}$$

In addition to pheromone, trail table must be initialized. This trail is used as heuristic information, so that the ants can choose the shortest path between the options of the next nodes. Trail is initialized by using the equation as follows:

$$Trail_{ij} = \frac{1}{Cost_{ij}} \tag{5}$$

3.2 The Selection of Next Node

At the time of construction, an ant from ant colony will look for the next node by checking the list of nodes that are allowed to pass (has never been passed), from the nodes allowed to pass, it is collected the value of trail and pheromone from each node as divider probability,

$$denom = \sum_{j \in allowed} \left(Pheromone_{ij}^{alpha} \right)\left(Trail_{ij}^{alpha} \right) \tag{6}$$

This denominator is used to find the probability of a node to be selected, using the following equation

$$prob_{ij} = \frac{(Pheromone_{ij}^{alpha})(Trail_{ij}^{alpha})}{denom} \tag{7}$$

After obtaining the probability value of each allowed node (allowed), then sistem selects the next node. System will generate a random value from 0 to 0.99. The probability value of each node will be compared to the generated random value. If the value $Prob_{ij}$ is greater than the random value, then j is selected as next node, meanwhile, current node is replaced by j, and j is removed from allowed table, then cost from i to j is stored in total cost and j is entered into tabu. if the probability value is less than the random value, then the random value will be subtracted with the node's probability value compared, then it is compared again with the probability value of the other node until the probability value is greater than the random value, and so on until the node has been selected all. After the ant has visited all the nodes, the total cost of the ants that have made the path will be stored. If the cost value is smaller than the best total cost, then the value of total cost is replaced by the total cost value of the ant.

3.3 Updating pheromone
Each iteration will generate a route of a certain amount of ant colony, and the route will be selected with the lowest cost value. Furthetmore, each node path of the route that has been raised by each ant will have an updated pheromon value. Hence, the node that is often passed by its pheromone will be thickened, and the possibility of the node to be re-selected will be higher. The pheromone will be updated by using the equation as follows,

$$PheroDelta_{ij} = \beta(pheromone_{ij}) + \frac{1}{cost_{ij}} \qquad (8)$$

The nodes that has not passed by pheromone will be multiplied by evaporation rate.

4. Our Proposed Model
Generally, this system as shown in Figure 1 consists of two main processes, the first is the process of choosing the desired tourist attractions such as natural attraction tourism, traditional restaurants, shopping places, etc. The system recommends an optimal travel route schedule (travel itinerary) based on user needs, which is represented in degree of interest (DOI). There are 3 criteria of need given, 1) route with as many tourist attractions as possible, 2) routes with popular attractions, 3) routes that minimize the budget. The system will generate a solution for scheduling based on the user input, using ant-collony optimization, and then implement its results into google map. Figure 1 shows the implementation of our system. Degree of Interest (DOI) is very influential on the solution generated by the algorithm. The value of DOI is within the interval [0,1].
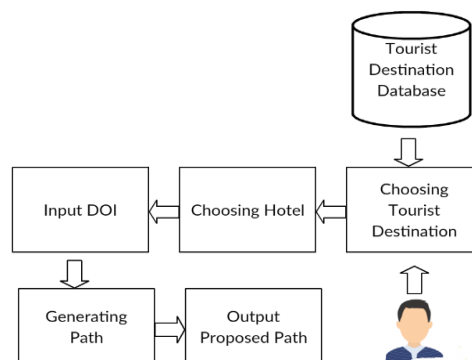

Figure 1. System implementation

To accommodate that multi criteria case, we use MAUT concept. To determine the cost of an edge, we use equation 2. There are three constraints for generating itinerary as follows,

a. The trip begins (start time) at 08.00 and must have returned to the hotel before 20.00 (end of the day)
b. The maximum limit of scheduling is 3 days
c. There is no user limitation in choosing a tourist attraction

Time is the principal constraint in the created system, because time affects the combination of a generated solution. In a tourist attraction, there is open and closing hours table, time travel table to other tourist attractions and the average time travelers visit the place (average time). Each ant has a variable currenttime. This variable is initialized with the start time. Once the ant visits nodes, the currenttime of each ant will be added as the amount of travel time to the node and added with the average value of the average time at the node.

This variable currenttime is used to check for a node, whether fulfill its constraint or not (open hours, closing hours and End of The Day). Variables allowed (the next nodes candidate) is being checked whether meet the constraint, when it is fulfilled, then it will be entered into the new variable tempallowed. This variable tempallowed is used to replace the allowable variable for probability. Algorithm 2 shows the steps for filtering allowed nodes.

Algorithm 2. FilteringAllowedNode(ant,sch)
input: ant, table of open time and close time (schedule)
1: i ← 0
1: for (i < length(ant.allowed)):
3: if (ant.currenttime > sch['open'][ant.allowed[i]])
   and (ant.currenttime < sch['close'][ant.allowed[i]])
   and (ant.currentime) < endOftheDay):
4: tempallowed[]←ant.allowed[i]
5: return tempallowed

5. The Experiments
5.1 Evaluation of Running Time

We did a series of tests to get the best parameters for the ant algorithm. The parameters obtained are, α=0.7, β=4, ρ=0.8, q=25. Parameter α is pheromone factor, which is used to control pheromone values. Parameter β is the heuristic factor of a parameter used to adjust the heuristic value that affects the chance of a node with the smallest cost selected to be the solution. Parameter ρ is the evaporation rate. If the ant does not pass through a node then the pheromone will vaporize amounted ρ. Parameter q or quality is a regulating value for the number of pheromones released when the ant passes through the node. After obtaining the best parameters, a test is performed to obtain a comparison of the number of selected nodes with the number of user-entered nodes, the ratio of running time to the number of nodes and the ratio of the total time totals to the user-entered nodes.
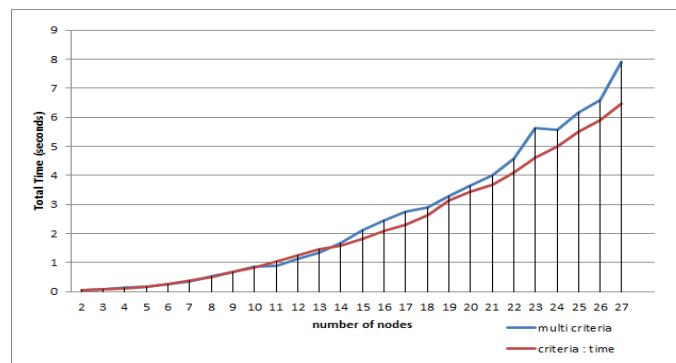

Figure 2. Running time test results versus the number of nodes entered by user

Running time is the time needed for the algorithm to execute the process to get the solution. Based on the Figure 2, there is a little difference in running time between ant-collony by using time-based and multi-criteria-based. When the number of nodes entered more than 14,

ant-collony using multi-criteria-based takes more time about less than 1 second. This is because when we use multi criteria, the processes will be more complex due to normalizing and calculating the value of cost that apply MAUT.

5.2 The evaluation of the number of selected nodes

The number of selected nodes is the total nodes that becomes the solution of the scheduling for 3 days of scheduling. Figure 3 shows that, if the user enters 2 to 8 nodes, the number of nodes that become TSP solution will be equal to the number of nodes the user enters. After exceeding 8 nodes, the solution node will not be equal to the number of nodes we input, because there is a limitation in the number of days. If we use multi-criteria-based, it can be seen the number of nodes that become solutions within the range of 11 nodes to 17 nodes, it is greater than the results obtained by using time-based. This is because when we apply MAUT, the opportunity of ants to generate different solutions will be high, thus leading to the generation of more diverse solutions.
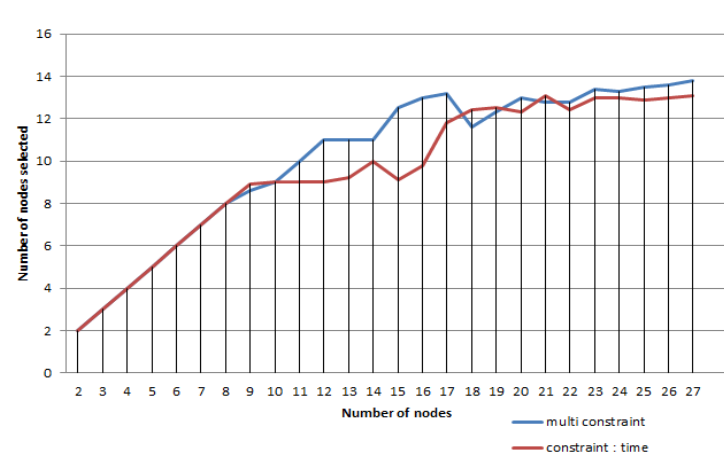

Figure 3. The test result of the number of selected nodes versus the number of nodes entered

5.3 Evaluation of the total days required

In this case, each algorithm is tested to determine how many days it takes to make a schedule of the number of different nodes. Here are the results of the tests performed. Based on Figure 5, we can notice that, if the user enters between 4 to 7 nodes, the algorithm will generate the solution for a total of two days. The differences occur when the number of nodes entered between 7 to 9 nodes between ant-collons using multi-criteria-based requires less time-based than time-based. It happens because when using multi constraint the combination of scheduling solutions is more diverse so results are better.
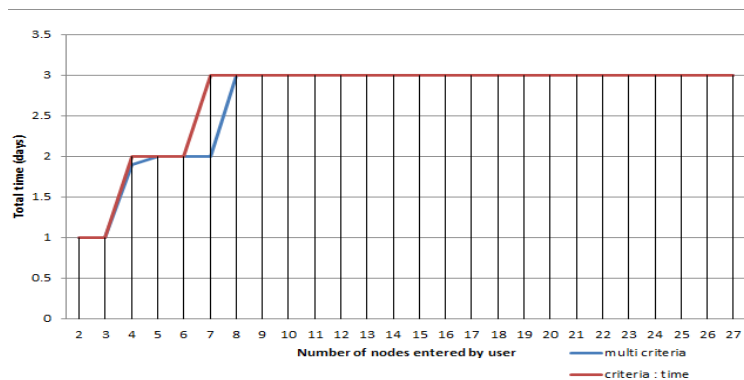
Figure 4. Total time test results towards the number of user-entered nodes

6. The Overview of Interaction

The first interaction is choosing the tourist attractions provided by sistem. Every tourist place has attributes such as tourist attraction name, address, telephone number and rating. User is free to select some places of interest, with no maximum limit as shown in Figure 6. Then, user is welcome to choose one of several hotels provided. Users can only choose one hotel only once in generating scheduling as shown in Figure 7.
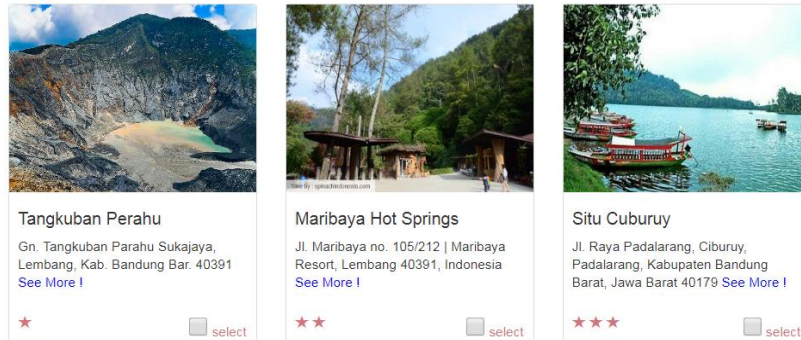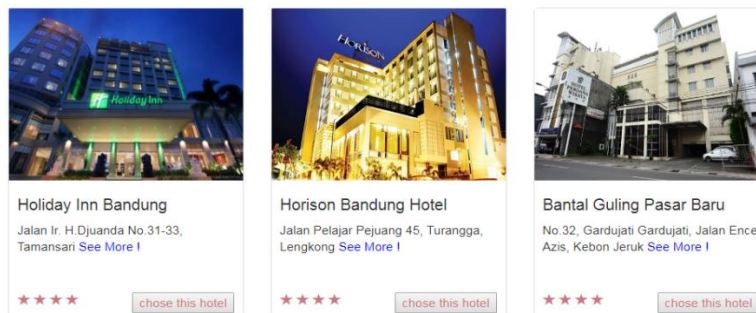

Figure 6. Choosing Tourist Attractions

Figure 7. Choosing Hotel

Furthermore, users are allowed to set the degree of Interest (DOI) as needed, for example, when the user wants to get the results of scheduling with cheap places, the user can set DOI "cheapest tour priority" and reduce other DOI as shown in Figure 8. The result of this scheduling is a sequence to visit tourist attractions for a maximum of three days (travel Itinerary), as shown by Figure 9. For each day, tour starts from selected hotels, and it can not be different hotels for each day. This visit sequence ends with an estimation of user when they arrive at the hotel.
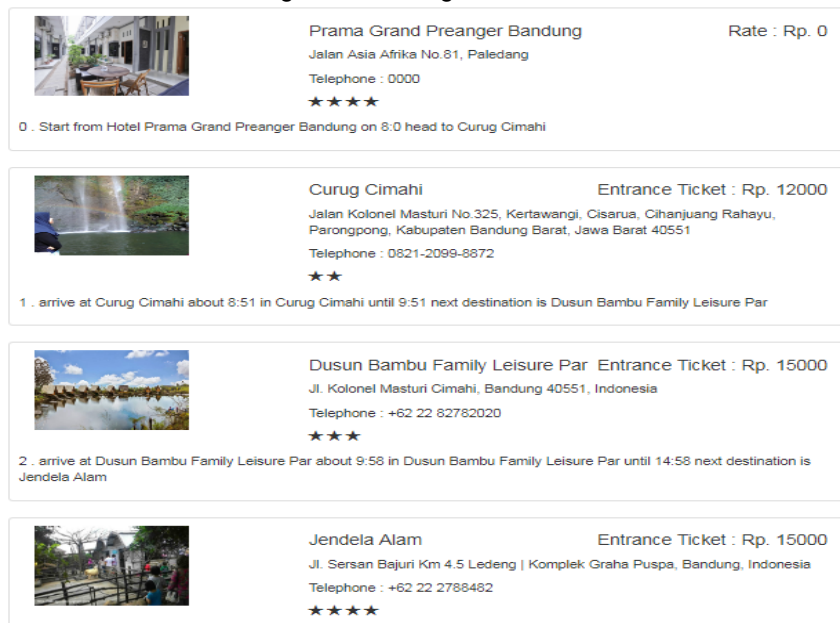
Figure.8 Set degree of Interest

Figure 9. Travel Itinerary

7. Conclusion and Future Work

Based on experiments, we notice that ant-collony is suitable for the generation of tourist route itinerary, because the ant colony checks every node whether fulfills the constraint or not. The running time of multi-criteria-based itinerary is not significantly different from time-based itinerary. In addition, the number of tourist attractions in the itinerary is more than time-based itinerary, as the combination of solutions from each ant becomes more diverse. In addition, the results obtained to get a schedule with a total time of 2 days, the user is suggested to enter the number of nodes around 8-11, for three days of scheduling, the user is suggested to enter about 16-20 nodes. Although it takes longer running time for about seconds, the multi-criteria approach makes the selected node test results and the total time required is better than time-based approach.

This research is a part of a tourism recommender system that we will develop. The system will provide recommendations on tourist destinations based on their needs, and users are given the opportunity to choose some of the recommended tourist destinations. Then the system will recommend a tour route based on the selected tourist destinations. The interaction of the recommender system, adopted from the interaction model introduced by Baizal et al. [15].

Acknowledgments

References
[1]   Vihikan W.O, Putra I.K.G.D. and Dharmaadi, I.P.A., 2017. Foreign Tourist Arrivals Forecasting Using Recurrent Neural Network Backpropagation through Time. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*. 2017: *15*(3): 1257-1264.
[2]   Lee C, Chang Y, Wang M. Ontological Recommendation Multi-Agent for Tainan City Travel. *Expert Systems and Applications*. 2009: 36(3): 6740–6753
[3]   Le T. Q. Pishva D. *An Innovative Tour Recommendation System Fortourists in Japan*. In 2016 18th International Conference on Advanced Communication Technology (ICACT). 2016: 717-729.
[4]   Adrissono L, Goy A, Petrone G, Segnan M, Torasso P. INTRIGUE: Personalized Recommendation of Tourist Attractions for Desktop and Handheld Devices. *Applied Artificial Intelligence,* 2003: 17: 687-714.

[5]   Vansteenwegen P, Souffriau W, Berghe G V, Oudheusden D. The City Trip Planner: An Expert System for Tourists. *Expert Systems with Applications*. 2011: 38(6):6540–6546

[6]   Rodrıguez B, Molina J, Perez F, Caballero R. Interactive Design of Personalised Tourism Routes.*Tourism Management*. 2012: 33(4): 926–940.

[7]   Yang L, Xiao-bo Y. Optimizing Fuzzy Petri Nets by Using an Improved Ant Colony Algorithm. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*. 2016: 14(2A): 129-137.

[8]   Yang X. *Nature-Inspired Metaheuristic Algorithms*. Luniver press. 2010.

[9]   Świątnicki Z. *Application of Ant Colony Optimization Algorithms for Transportation Problems Using the Example of the Travelling Salesman Problem*. In 4th International Conference on Advanced Logistics and Transport (ICALT), IEEE. 2015.

[10]   Liu X, Yi H, Ni Z. Application of Ant Colony Optimization Algorithm in Process Planning Optimization. *Journal of Intelligent Manufacturing*. 2013: (24).1: 1-13.

[11]   Scḧafer R. *Rules For Using Multi-Attribute Utility Theory for Estimatinga User's Interests*. In 19th Workshop Adaptivit ̈at und Benutzermodellie-rung in Interaktiven Software systemen. 2001: 8–10.

[12]   Bichler M, Kalagnanam J. Configurable Offers and Winner Determination in Multi-Attribute Auctions. *European Journal of Operational Research*. 2005:160(2): 380-394.

[13]   Torrance G.W, Boyle M H., Horwood S P. Application of Multi-Attribute Utility Theory to Measure Social Preferences for Health States. *Operations Research*. 1982: 30(6): 1043-1069.

[14]   Dorigo M, Gambardella L M. Ant Colony System: a Co-Operative Learning Approach to The Traveling Salesman Problem. *IEEE Transactions On Evolutionary Computation*. 1997: 1(1):53–66.

[15]   Baizal Z. A, Widyantoro D H, and Maulidevi N U. Factors Influencing User's Adoption of Conversational Recommender System Based on Product Functional Requirements. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*. 2016: 14(4): 1575-1585.