

Ant Colony Optimization on Crowdsourced Delivery Trip Consolidation

Victor Paskalathis*¹, Azhari SN.²

¹Program Studi S2 Ilmu Komputer, FMIPA UGM, Yogyakarta, Indonesia

²Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta, Indonesia

e-mail: *vic.easter@gmail.com, arisn.softcomp@gmail.com

Abstrak

Pengiriman barang berbasis urun daya umumnya dilakukan melalui pengiriman langsung, yakni dengan mengutus kurir terdekat dari lokasi asal. Jarak tempuh keseluruhan dapat dikurangi melalui pengambilan dan pengiriman jamak, yakni dengan memperbanyak jumlah permintaan yang dapat dipenuhi dalam satu perjalanan.

Penelitian ini mengimplementasikan algoritma eksak untuk menyelesaikan masalah konsolidasi hingga tiga permintaan dalam satu perjalanan. Algoritma serakah digunakan untuk menyusun rute awal berdasarkan penghematan jarak tertinggi. Hasilnya kemudian dioptimasi menggunakan algoritma semut. Dalam penelitian ini empat skenario dibandingkan, yakni skenario pengiriman langsung dan tiga skenario pengambilan dan pengiriman jamak. Skenario ini mencakup konsolidasi dua pengiriman, tiga pengiriman dan tiga pengiriman yang dioptimasi menggunakan algoritma semut. Empat parameter digunakan untuk mengevaluasi tiap skenario menggunakan analisis hirarki proses (AHP). Parameter tersebut adalah jumlah perjalanan, jarak tempuh total, waktu tempuh total dan pertimbangan keamanan.

Studi kasus yang digunakan adalah wilayah operasional Yogyakarta selama sehari penuh. Rute pengiriman yang dioptimasi menggunakan algoritma semut menunjukkan bahwa 178 permintaan dapat dipenuhi dalam 94 perjalanan. Rute ini juga memberikan penghematan jarak tempuh sebesar 20% dan waktu tempuh sebesar 15%. Hasil evaluasi menggunakan AHP menunjukkan skenario algoritma semut sebagai skenario terbaik..

Kata kunci—algoritma semut, masalah pengambilan dan pengiriman, penghematan tertinggi, urun daya, konsolidasi perjalanan

Abstract

Common practice in crowdsourced delivery services is through direct delivery. That is by dispatching direct trip to a driver nearby the origin location. The total distance can be reduced through multiple pickup and delivery by increasing the number of requests in a trip.

The research implements exact algorithm to solve the consolidation problem with up to 3 requests in a trip. Greedy heuristic is performed to construct initial route based on highest savings. The result is then optimized using Ant Colony Optimization (ACO). Four scenarios are compared. A direct delivery scenarios and three multiple pickup and delivery scenarios. These include 2-consolidated delivery, 3-consolidated delivery, and 3-consolidated delivery optimized with ACO. Four parameters are used to evaluate using Analytical Hierarchical Process (AHP). These include the number of trips, total distance, total duration, and security concerns.

The case study is based on Yogyakarta area for a whole day. The final route optimized with ACO shows 178 requests can be completed in 94 trips. Compared to direct delivery, consolidation can provides savings up to 20% in distance and 14% in duration. The evaluation result using AHP shows that ACO scenario is the best scenario.

Keywords—Ant Colony Optimization, Pickup and Delivery Problem, highest savings, crowdsourced, trip consolidation

1. INTRODUCTION

Delivery performance can be evaluated by its effectiveness and efficiency. Effectiveness can be illustrated through direct delivery. This is the current practice of crowdsourced delivery services [1]. Delivery is done point to point, from an origin to a destination and this is done directly. In this scenario, the goods can reach the destination fastest in a single case. There is an issue in this method. Sometimes there is another request nearby or along the route which cannot be accepted since the driver already took the request. Another problem may also occur in collection or distribution case, where the request is to collect goods from multiple different origins and deliver to a single origin or vice versa.

The above condition can be anticipated with multiple pickup and delivery or also known as consolidated delivery [2]. In this scenario, the efficiency can be maximized. Delivery is done by consolidating requests in the same direction so the vehicle capacity can be better utilized. This will also reduce the overall mileage, time on the road, and numbers of trip generated. The outcomes are lower carbon emission, traffic generation, and reduced travel cost.

A simple static instance is shown in **Fig. 1** The multiple pickup and delivery can provide 25% distance savings and 30% duration savings while still maintaining 1-hour guaranteed delivery. This is done with only 2 drivers compared to 5 drivers using direct delivery. By increasing the number of requests in a trip, the distance savings can be maximized. This is quite an opportunity. Current conventional courier services are using next-day delivery and same-day courier costs more expensive than crowdsourced delivery services.

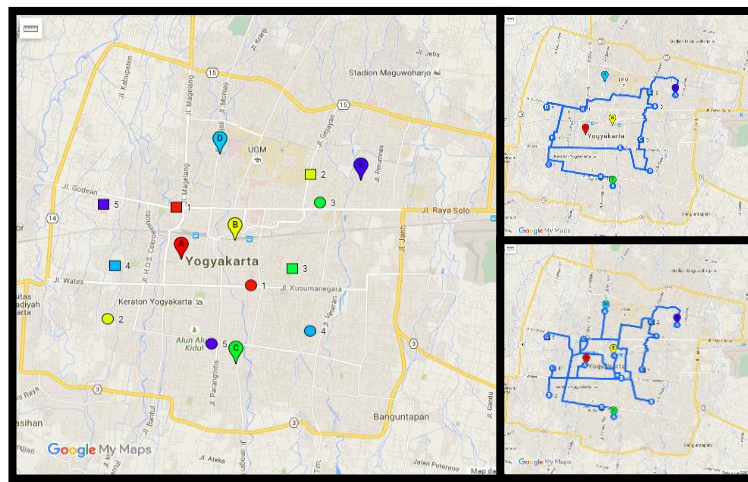


Figure 1 Problem illustration (left) and solution using multiple pickup and delivery (top right) and direct delivery (bottom right)

Problems in transporting goods on delivery services can be described as *Pickup and Delivery Problem* (PDP), a constrained version of *Traveling Salesman Problem* (TSP). The problems are NP-Hard in nature [3] and two classes of algorithms are available for the solution: exact and approximate algorithms [4].

Exact algorithm guaranteed to find the optimal solution, however, in the worst case it requires exponential time to find the optimum. As the problem size increase, it quickly becomes unfeasible. Approximate algorithms seek to obtain high-quality solutions at relatively low computational cost without being able to guarantee the optimality of solutions. One of the modern approaches is by using metaheuristics, such as Ant Colony Optimization.

Ant Colony Optimization is a population-based metaheuristic for the approximate solution of difficult optimization problems which inspired by ant foraging behavior [5]. Their way of foraging enables ants to find shortest paths between food sources and their nest. Initially, ants explore the area surrounding their nest randomly. As soon as an ant finds a source of food,

it evaluates quantity and quality of the food and carries some of this food to the nest. During the return trip, the ant lays chemical pheromone trail on the ground. The quantity of pheromone deposited will guide other ants to the food source.

The problem to be solved in this research is to optimize current crowdsourced delivery practice by providing additional scenarios in delivery scheme. Several multiple pickup and delivery scenarios are developed to be compared with current direct delivery. The system is required to consolidate multiple requests into a trip, construct a route consists of trips to fulfill all requests, and provide the best scenario to consider.

Three objectives are presented. The first objective is to develop several scenarios in crowdsourced delivery services based on multiple pickup and delivery. The second objective is to implement Ant Colony Optimization in route optimization. The last objective is to evaluate the scenario using Analytical Hierarchical Process and provide the best scenario to consider.

Study on *pickup and delivery problem* (PDP) has been done since decades ago, as it represents the general transportation problem with precedence constraint in term of origin and destination. One of the earliest studies of PDP was conducted by [6] in solving single and multiple vehicle PDP using various exact and heuristic algorithms. The most general PDP is constrained by capacity, time windows, quality of service, operational considerations and the precedence requirement.

Presented metaheuristic solution of PDP with time windows for city courier [7]. The research addresses both online and offline requests in the form of PDP and dynamic PDP with time windows. Dynamic PDP can be seen as a static problem which occurs sequentially. Using 2-stage algorithms, the research implements *Insertion Heuristic* for tour construction and improved by Tabu Search. The route update is carried on the dynamic problem when a new request arrives.

Early research in PDP using Ant Colony Optimization (ACO) presented by [8] to optimize the total costs associated with the pickup and delivery of full truckloads under time window constraints in a hub network. A thorough technical analysis of the ACO was done by comparing different pheromone decoding schemes, different visibility information, and various population sizes. The study also proposes a post-optimization technique to improve the solutions. The results show that appropriate data structures significantly improve the solution quality.

Compares *Improved Ant Colony System* (I-ACS) algorithm towards *Nearest Neighbor*, *Nearest Neighbor + Insertion* on both dynamic and static VRP and PDP problem on city and inter-city courier [9][10]. Node insertion was used to determine route and number of appropriate vehicle on dynamic PDP. Shortest path calculated using Dijkstra since the distance uses actual city road data.

Analytical Hierarchical Process (AHP) [11] implemented in this research to evaluate the scenarios since it is easy to perform and based on pairwise comparisons to maintain consistency of each pairwise values. [12] have successfully implement AHP to compare results from various VRP algorithm.

2. RESEARCH METHOD

2.1 Model Formulation

The objective function in this research is to maximize the distance savings. The distance savings are the difference between sum of direct trip of each request and total distance of consolidated trip. Subject to:

1. Flow constraint, ensures a request only assigned to one vehicle.
2. Flow conservation, ensures a vehicle only enters or leaves a location l if it is an origin or a destination of a transport request assigned to that vehicle.
3. Consolidation limit constraint, ensures that a vehicle only assigned to a maximum n number of requests in a trip.

4. Schedule compatibility, serves as preliminary check for maximum time range between requests before driver is dispatched.
5. Pairing constraint, ensures that both origin and destination of a request must be served by the same vehicle.
6. Precedence constraint, ensures each pickup location visited before the delivery location.
7. Chaining constraint, disallows direct trip at the start or the end of the trip. request in consolidated trip shall not be independent, since direct trip has no increase in distance savings. Such trip should be split.
8. Savings constraint, ensures the consolidated trip will provide positive distance savings. Distance of consolidated trip must be equal or lesser than sum of direct trip of each request.
9. Time windows and service time constraint, serve as quality of service constraint. The maximum service time for pickup and time windows for delivery is 1 hour and 3 hours since request created.
10. Best alternate constraint, if there are several trips serving same requests, the selected trip is the one which provide highest savings. Trip distance savings is the sum of all direct trip of each request minus the consolidated trip.

2.2 System Process

Fig. 2 shows the system process. Raw datasets are preprocessed to produce distance and duration matrix off the system. Both data then used as inputs. The first step in the system tries to generate all possible route based on the constraint using the exact algorithm. The second step is to find a route with highest savings using greedy the algorithm. The results are used as Scenario 1, 2, 3 for direct, c=2 and c=3 respectively.

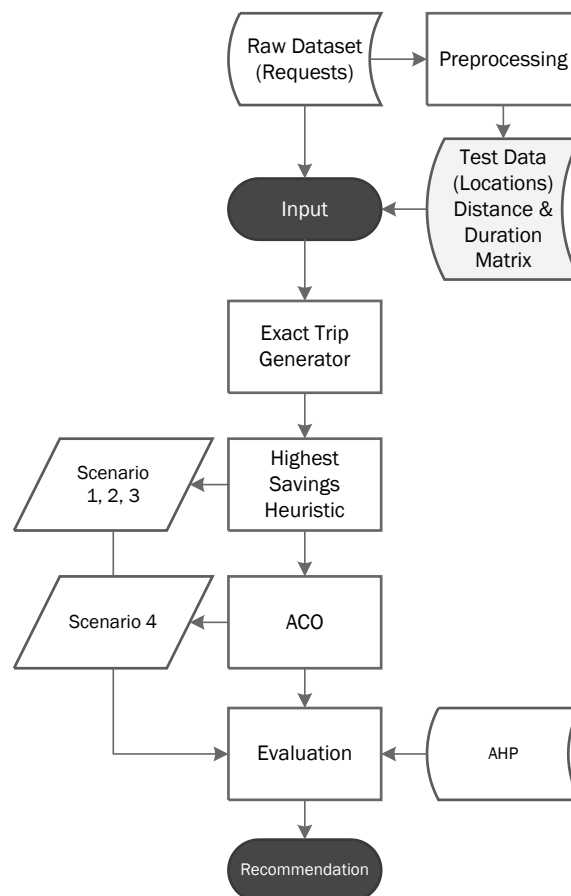


Figure 2 System flowchart

The third step uses the ACO [13] to optimize the result based on the previous attempt on $c=3$. The result is used as Scenario 4. The last step is the evaluation. This is done by implementing AHP method based on four parameters. The final output is the recommendation on which scenario is better to be implemented.

2.3 Exact Trip Generator

The exact trip generator solves two of three problems in PDP; routing and scheduling. This is done through the exact algorithm by applying PDP constraint. The trip generated by static routing has total pickup based on consolidation or in $c=3$ scenario, it has either 2, 4, or maximum 6 stops. Trip with highest savings become the valid trip to serve unique requests. Fig. 3 shows the detailed process in this step.

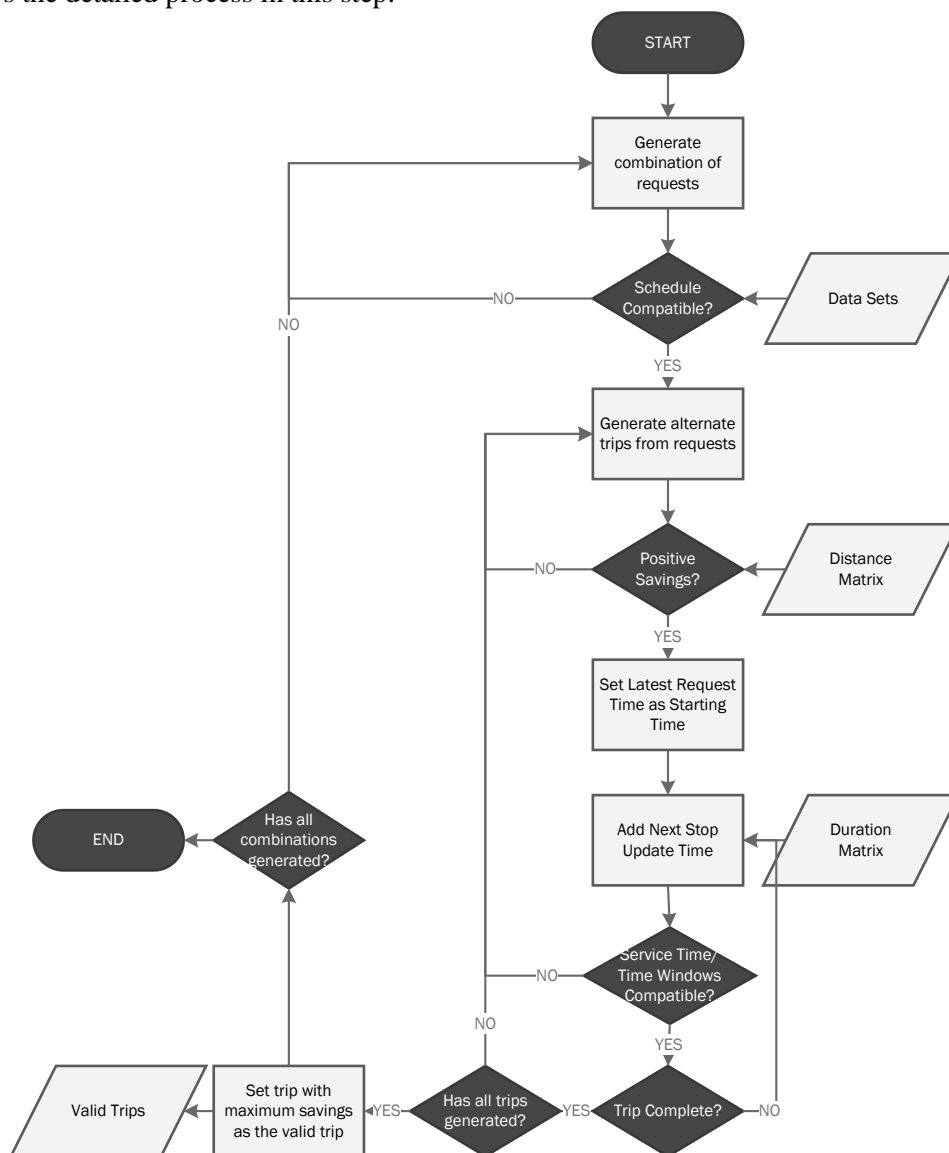


Figure 3 Exact Trip Generator

The approach is quite similar with Clarke-Wright Savings Algorithm [14], with two key differences. The trip doesn't start from a depot, the trip also not required to return to the depot. In this case, there might be negative distance savings.

2.4 Highest Savings Heuristic

Highest savings heuristic construct a route which serves all of the requests. This is assignment problem, through finding the combination of previously generated trips which has highest savings to serve all requests. **Fig. 4** describe the process in the current step.

The system utilizes greedy algorithm based on highest savings. The approach is similar to nearest neighbor, instead of shortest distance, largest distance savings used. The valid trips sorted in descending based on savings. The trip with largest savings is assigned to route until there's no more valid consolidation left. The remaining requests are fulfilled with direct delivery.

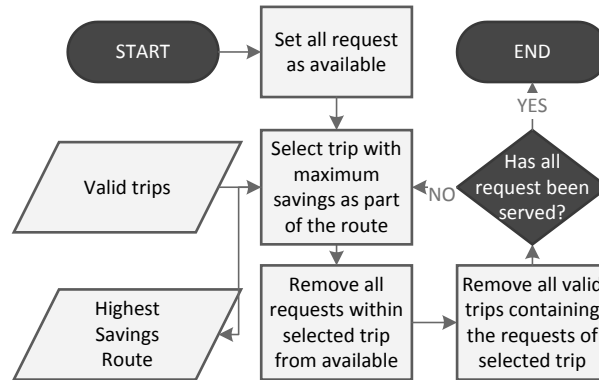


Figure 4 Highest savings

2.5 Ant Colony Optimization

The optimization process based on ACO can be seen in **Fig. 5**. This can be divided into 3 main stages. First is the initialization of parameter, followed by solution construction, and pheromone updates and local optimization. After the maximum iterations, it returns the best-found route. This research is using ACO variant known as Ant Colony System (ACS) [15].

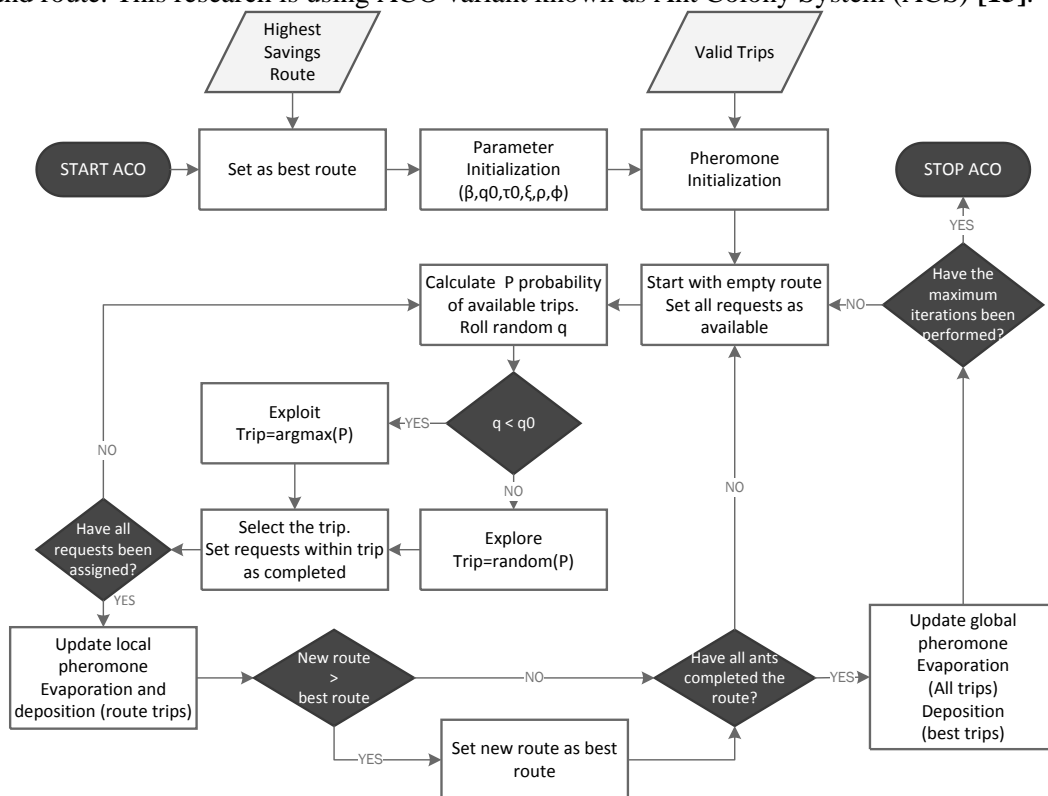


Figure 5 Ant Colony Optimization

The research modifies several factors from standard ACS:

1. $\tau_0 = \frac{s^{ls}}{n}$; where s^{ls} is the savings generated from highest savings heuristic and n the number of trips. Required to accommodate savings maximization
2. $\eta_{i,j} = s_{i,j}$ is also required to accommodate savings maximization.
3. ϕ phi parameter in pheromone initialization, with $\phi = (0, 1)$. This parameter is useful to prevent the optimization stuck on local optima. The downside is that it will take more loops to find better results.

2.6 AHP Evaluation

This research doesn't end with the best route, but also a recommendation towards the business development plan. This is completed using AHP with overall performance as the goal. The basic structure of a decision problem is a hierarchy consisting of 3 levels. The goal of the decision at the top level. The second levels are parameters used. The third levels are alternatives to be evaluated. Fig. 6 shows the hierarchy.

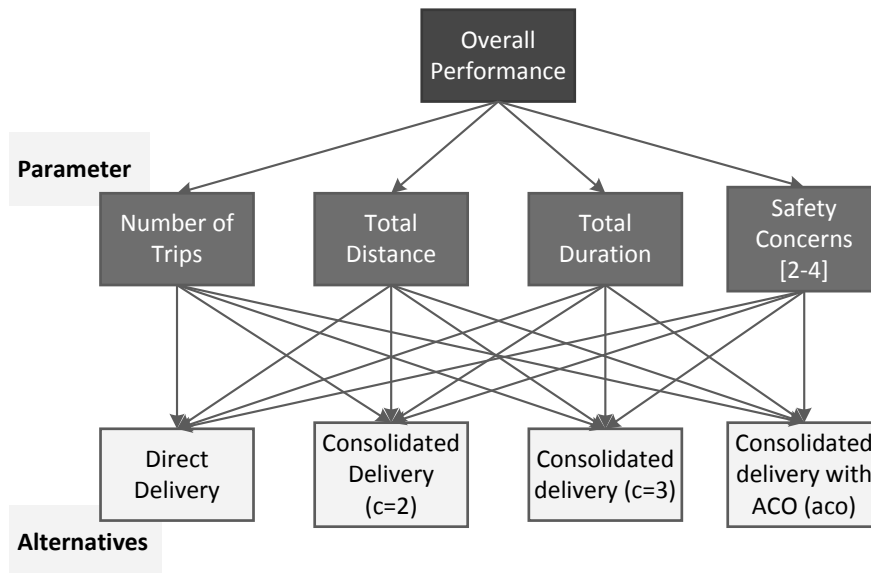


Figure 6 AHP Hierarchy

2.5.1 Parameter weights

Pairwise value for each parameter weight are used, ranging from 0-2.5. Since there are equal parameters and scenarios used, which is 4, then Random Index in use is 0.9. The result of the weight can be seen in Table 1. The CR value is 0.016 which is below 0.1. Based on the EV, the parameter which has the highest weight is the number of trips, followed by total distance and security concerns, with total duration least considered.

Table 1 Weight EV

Weight	Trip	distance	Duration	Security	Ev
Trip	1.00	1.50	2.50	1.50	0.366
Distance	0.67	1.00	2.00	1.00	0.257
Duration	0.40	0.50	1.00	1.00	0.160
Security	0.67	1.00	1.00	1.00	0.217

2.5.2 Scenario rating

Rating method for each scenario based on parameters are shown below:

1. Number of Trips: to be reduced, lower value preferred.
2. Total distance: to be reduced, lower value preferred.
3. Total duration: to be reduced, lower value preferred.
4. Security concerns: implemented with scoring from 2-4 with higher value preferred as below:
 - 1) direct → score=4
 - 2) c=2 → score=3
 - 3) c=3, aco → score=2

3. RESULT AND DISCUSSION

Table 2 shows an overview of each scenario result. In general, increase in the number of consolidation yields reduction in the number of trips, total distance, and total duration. Savings value is compared with *direct* data. *Aco* is used to compare since it provides highest savings.

Table 2 Summary of figures

	<i>direct</i>	<i>c=2</i>	<i>c=3</i>	<i>Aco</i>
<i>Trips</i>	178	110	97	94
<i>Direct Trips</i>	178	42	47	40
<i>Distance</i>	1,309,800	1,091,971	1,056,102	1,050,761
<i>Distance savings</i>	0	217,829	253,698	259,039
<i>Duration</i>	200,197	180,116	173,590	172,984
<i>Duration savings</i>	0	20,081	26,607	27,213

Overall the routing results show promising savings can be done by implementing multiple pickup and delivery. The number of trips reduced from 178 to 94 which is 47% reduction. The total distance is reduced by 20% and duration is reduced by 14%. The consolidation still leaves several trips to be completed with direct delivery, or 42% in *aco* scenario. There are also 6 trips that have no valid consolidation and completed with this method.

Fig. 7 shows the number of trips each hour between scenarios. In peak hour between 15:00 to 15:59, the consolidation able to reduce the number of trips more than 50% number of trips. This shows that the higher number of requests per hour, the higher chance of requests can be consolidated.

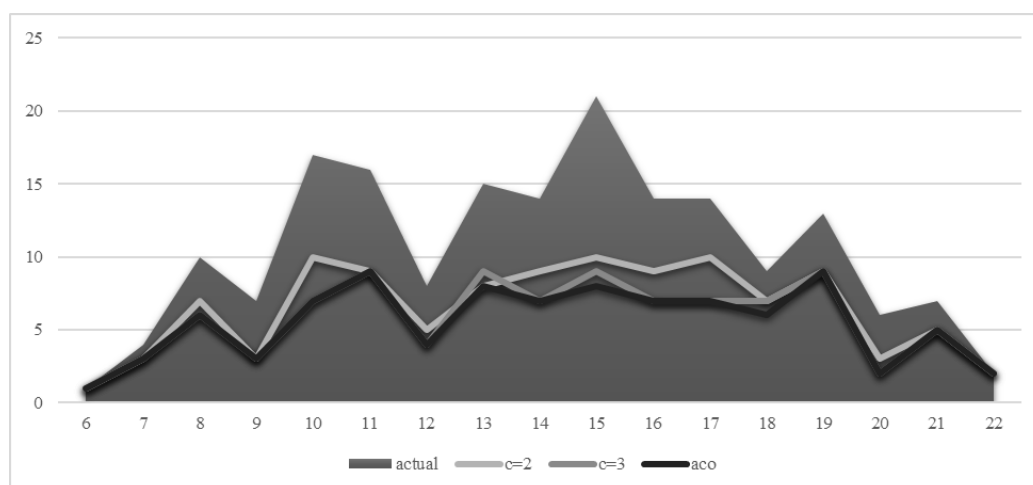


Figure 7 Trip per hour

Fig. 8 shows how the ACO iterations producing distance savings result. The best route occurs in 5th iteration 10th ant with 259,039 distance savings.

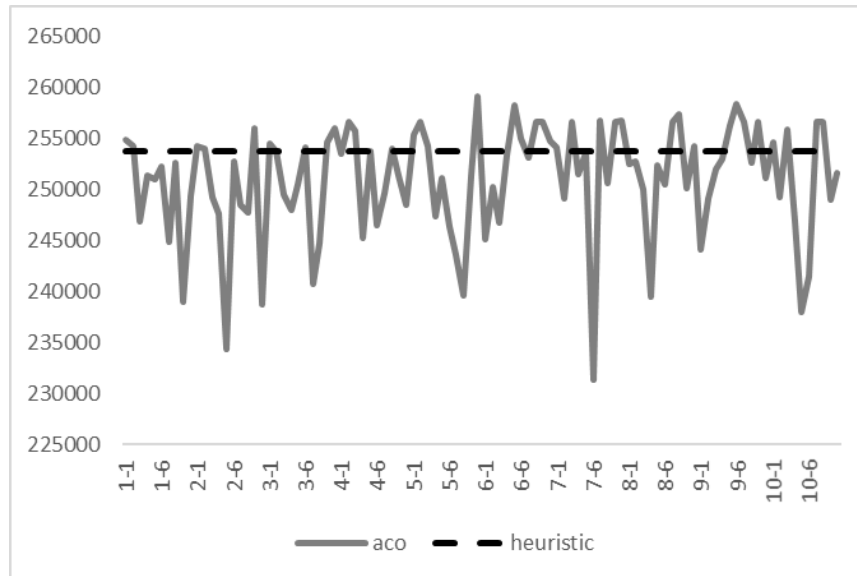


Figure 8 ACO iterations

Table 3 shows final AHP results. The best scenario is the ACO scenario, with slight difference to $c=2$ scenario. Higher consolidation with proper optimization provides a better result with downside on the security issue. In a broader scope, consolidated delivery result in better score compared to instant delivery.

Delivery	Trip	distance	duration	Security	Score	rank
Weight	0.366	0.257	0.160	0.217		
direct	0.158	0.213	0.226	0.364	0.228	4
$c=2$	0.255	0.256	0.251	0.273	0.258	3
$c=3$	0.289	0.265	0.261	0.182	0.255	2
aco	0.298	0.266	0.262	0.182	0.259	1

4. CONCLUSION

Based on the research, there are several things to be concluded:

1. The research has successfully developed 4 scenarios to be compared. The direct delivery, 2-consolidated delivery, 3-consolidated delivery and 3-consolidated delivery optimized with ACO.
2. The research has successfully optimized the route using ACO. It shows improvements in considerable time from heuristic highest savings. It also shows that ACO able to work combined with both exact and heuristic algorithm to produce a satisfactory result.
3. The research has evaluated all scenarios using AHP. Using 4 parameters, the scenario with the highest score is 3-consolidated delivery optimized with ACO and therefore recommended to be implemented.
4. Overall the multiple pickup and delivery result is quite promising. 3-consolidated delivery optimized with ACO results in 20% distance reduction and 14% duration reduction compared to direct delivery.

5. FUTURE WORKS

There are several suggestions can be given from this research:

1. Larger problem size, the number of request in Yogyakarta Area is quite small compared to Jakarta. The algorithm should be able to handle at least thousands of requests.
2. Dynamic routing can be developed to compare with current static routing. It may be implemented in the main system to allows dynamic consolidation in real time and handle cancellation.
3. Parallelization through multicore/multiprocessing by running each ant in parallel instead of the current sequential system to decrease running time.

REFERENCES

- [1] USPS OIG, 2014, *Using the 'Crowd' to Deliver Packages*, USPS OIG, Arlington.
- [2] Berling, P., & Eng-Larson, F., 2016, Pricing and Timing of Consolidated Deliveries in Presence of an Express Alternative - Financial and Environmental Analysis, *European Journal of Operational Research*, 250(2), pp. 590-601.
- [3] Savelsbergh, M.W.P, Sol, M., 1995, The General Pickup and Delivery Problem, *Transport Sci*, 29, pp. 17-29
- [4] Toth, P., & Vigo, D., 2002, *The Vehicle Routing Problem*, Society for Industrial and Applied Mathematics, Philadelphia.
- [5] Blum, C. and Roli, A., 2008, *Hybrid Metaheuristics: An Introduction*, Blum, C., Aguilera, M.J.B., Roli, A., Sampels, M. (eds.): *Hybrid Metaheuristics: An Emerging Approach to Optimization*, Springer-Verlag, Berlin.
- [6] Armstrong, G.R., 1981, *The Single and Multiple Vehicle Pickup and Delivery Problem: Exact and Heuristic Algorithms*, University of Tennessee, Knoxville.
- [7] Rani, F. K., 2008, Pengembangan Algoritma Heuristik untuk Penyelesaian Dynamic Pick Up and Delivery Problem with Time Windows (DPDTW) Pada Penyedia Jasa City – Courier, *Skripsi*, Fakultas Teknik Industri ITS, Surabaya.
- [8] Doerner, K., Hartl, R., Reimann, M., 2000, *Ant colony optimization applied to the pickup and delivery problem*, SFB Adaptive Information Systems and Modelling in Economics and Management Science, WU Vienna University of Economics and Business, Vienna.
- [9] Firmansyah, A., 2011, Algoritma Improved Ant Colony System untuk Menyelesaikan Dynamic Pick-Up and Delivery Problem with Time Windows Pada Penyedia Layanan Kurir Dalam Kota, *Skripsi*, Fakultas Teknik Industri ITS, Surabaya.
- [10] Gamayanti, N., 2009, Pengembangan Algoritma Heuristik Ant Colony System untuk Permasalahan Dynamic Vehicle Routing Problem dengan Time Window (DVRPTW) pada Proses Konsolidasi Penyedia Jasa Inter-City Courier, *Thesis*, Fakultas Teknik Industri ITS, Surabaya.
- [11] Saaty, T.L., & Vargas, L.G., 2012, *Models, Methods, Concepts & Applications of the Analytic Hierarchy Process*, 2nd ed. Springer, New York.
- [12] Cavar, I., Gold, H., & Caric, T., 2005, *Assessment of Heuristic Algorithms for Solving Real Capacitated Vehicle Routing Problems by Analytic Hierarchy Process*, Transportation Research Board, San Francisco.
- [13] Dorigo, M., & Stützle, T., 2004, *Ant Colony Optimization*, MIT Press, Cambridge.
- [14] Clarke, G. & Wright, J., 1964, Scheduling of Vehicles from a Central Depot to a Number of Delivery Points, *Operation Research*, Vol. 12, pp. 568-581.
- [15] Dorigo, M., & Gambardella, L., 1997, Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem, *IEEE Transactions on Evolutionary Computation*, 1(1), pp. 53-66.