An Architecture for Real-Time Location Recommendation for Field Data Collection

Aris Prawisudatama
School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Bandung, Indonesia
Email: soedomoto@gmail.com

I Gusti Bagus Baskara Nugraha
School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Bandung, Indonesia
Email: baskara@stei.itb.ac.id

***Abstract*—Field data collection is one of the main activities performed by the national statistical agencies in every country. Data collection activities have a similar workflow with Multi-Depot Vehicle Routing Problem (MDVRP). The use of MDVRP to generate pre-calculated routes resulted in total route costs with high standard deviation. The real-time mechanism by utilizing the publish/subscribe paradigm combined with Cooperative Coevolution Algorithms (CoEAs) is proposed to reduce the inequality (large variation) of the completion time. The test results show that routes produced by the combination of publish/subscribe paradigm and CoEAs are more prevalent in total route times among enumerators compared with the pre-calculated routes produced by CoEAs only.**

I. Introduction

Data collection is one of the main duties of a statistical agency in every country. There are two data collection methods that are generally used: primary and secondary data collection. The primary data collection requires a direct interview with the respondents, while the secondary data collection only compiles the data collected by other agencies. Related to this, the primary data collection involves an activity of allocating the enumerators into several census/survey areas. The allocation is usually conducted by assigning an equal number of locations on each enumerator.

The workflow of the data collection performs as follows:

1) Each enumerator gets a list of census/survey locations to visit.
2) The enumeration starts on one particular point. It could be the statistical office or the enumerator's house.
3) The enumerator moves to the first census/survey location and visits all respondents in that location.
4) After finishing the first census/survey area, the enumerator moves to the second census/survey location and continue doing so until all locations have been visited successfully.

This workflow is significantly similar to the Vehicle Routing Problem (VRP), specifically, the Multi-Depot Vehicle Routing Problem (MDVRP). The enumerators are analogous to vehicles, census/survey locations are comparable with customers, and the enumeration's starting point is corresponding with the depot.

However, although sharing similar workflow with MDVRP, the rough implementation of the MDVRP algorithm is not feasible for solving the allocation problems. This is because MDVRP needs the variable *service time*, which will not be available until after the enumerator finishes visiting the census/survey location. Therefore, this research aims to propose an integration of MDVRP with publish/subscribe mechanism to create a real-time location recommendation system for field data collection.

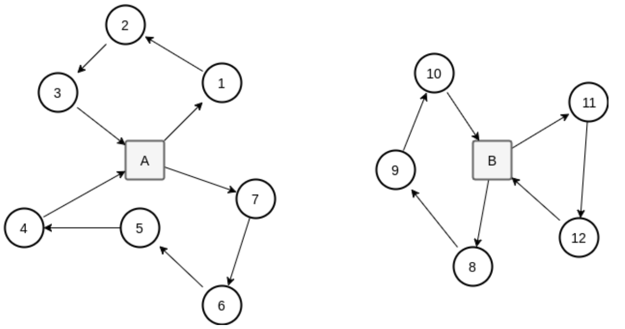

Fig. 1. An illustration for *Multi Depot* VRP

II. Related Works

A. Multi-Depot Vehicle Routing Problem (MDVRP)

VRP is an optimization problem that focused on the query: *'given a set of vehicles and a starting point (depot), find the best route to visit all customers'*. In cases there is more than one depot, VRP is known as MDVRP [1]. Figure 1 shows an example of an MDVRP solution that uses two depots and two routes that associate with each depot. Basically, a solution for MDVRP is a set of routes such that: (i) each route starts and ends at the same depot, (ii) each customer is only served once by one vehicle, (iii) the total demand on each route does not exceed vehicle's capacity, (iv) the maximum route time is satisfied, and (v) the total cost is minimized.

According to Renaud et al [2], the MDVRP can be formally described as follows:

1) Let $G = (V, A)$ be a complete graph, where V is the set of nodes and A is the set of arcs.
2) Partition the nodes into two subsets: the customers to serve V_C = {1,..., N} and the depots V_D = {N+1,..., $N + M$}, with $V_C \cup V_D = V$ and $V_C \cap V_D = \oslash$.

3) There is a non-negative cost c_{ij} associated with each arc $(i, j) \in A$.
4) The demand of each customer is d_i (there is no demand at the depot nodes).
5) There is also a fleet of K identical vehicles, each with capacity Q.
6) The service time at each customer i is t_i while the maximum route duration time is set to T.
7) A conversion factor w_{ij} might be needed to transform the cost c_{ij} into time units. In the classical MDVRP, however, the cost is similar with the time and distance units. Hence, w_{ij} = 1.

B. Evolution Algorithms (EAs)

Evolution Algorithms (EAs) is an optimization algorithm inspired by nature. EAs involve selection and reproduction processes to purify population from the solution candidates. According to Engelbrecht [3], the EAs lifecycle begins with randomly configuring the initial population. For each iteration, every individual in the population is evaluated using an objective function to find the solution candidates. Each solution candidate is labeled with a fitness value based on its evaluation result. These candidates will then reproduce and create the next generation. The whole process will be repeated using the new-generation individuals. The iteration will stop once it hits a certain time limit, which is 60 seconds.

In standard EAs, evolution is usually viewed as the population's attempts to adapt in a fixed physical environment. Furthermore, Coevolutionary algorithms (CoEAs), a development of EAs, realize that in natural evolution the physical environment is influenced by other independently-acting biological populations [3]. Based on the interaction of each species, CoEAs can be distinguished into two categories: competitive and cooperative [3]. In the competitive coevolution, each individual competes with other individuals in the same group. Meanwhile, the cooperative coevolution has its species mutually interacted or, at least, not harming each other.

C. Publish/Subscribe Paradigm

The publish/subscribe interaction is a communication method between client and server, which involves subscribers as clients. The subscribers are the parties who have interests in a certain event or the pattern of an event. These subscribers will get a notification from the publisher about the event they are interested in [4].

The basic model of the publish/subscribe system, as can be seen in Figure 2, depends on the event notification service that provides storage and management of subscription. This event service acts as a mediator between the publisher (the event's producer) and the subscriber (the event's consumer).

One of the benefits of using publish/subscribe mechanism is the loose coupling characteristic [4] between the publisher and the subscriber. The separation of information (decoupling) between publisher and subscriber can be divided into 3 (three) dimensions, which are:

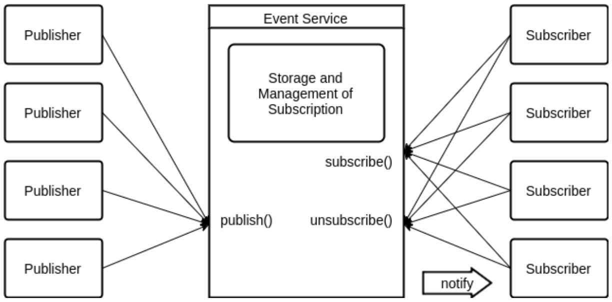

Fig. 2. Basic architecture in Publish/Subscribe Mechanism [4]

1) Space decoupling.
During the interaction, publisher and subscriber do not need to know each other. The publisher sends an event and subscriber indirectly receives the event through the event service. The publisher does not hold any reference towards the subscriber and vice versa.
2) Time decoupling.
The parties who are involved in the interactions do not have to interact in the same time. The publisher can send the event while in a disconnected state. Similarly, the subscriber can still receive the event, albeit disconnected.
3) Synchronizing decoupling.
Publisher will not be blocked during the event producing. The subscriber can still obtain information despite, at the same time, working on other tasks. Publisher and subscriber are not in the main flow, hence not interacting synchronously.

III. Proposed Solution

The drawback of using MDVRP for location recommendation is the absence of time service data, which is very important for computing the recommendation. Thus, MDVRP needs to be integrated with a real-time mechanism such as Web Service, Remote Procedure Call (RPC), message passing, and publish/subscribe mechanism [4].

In this research, the publish/subscribe mechanism is chosen because, in addition to its loose coupling characteristic, it is also suitable for an information-driven system [5]. Loosing coupling allows the publish/subscribe mechanism to work asynchronously, meaning the request and the reply do not have to be processed in sequence. The publisher and subscriber can commit offline publishing and subscription, respectively.

Figure 3 shows the system outline consists of 3 (three) main components: the publisher, the subscriber, and the message broker. The communication between publisher and subscriber is built upon their similarity in either event or topic. In this research, the subscriber's current location is chosen as the communication topic.

A. The Recommendation Publisher

VRP solver is implemented on the publisher's side. The publisher uses subscriber's current location as a topic to find the best solution/route. This route will be published to the

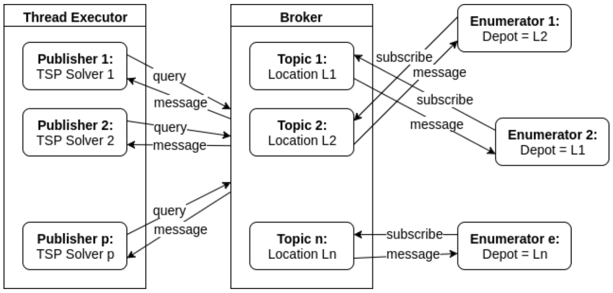

Fig. 3. The outline of the proposed system

subscriber via the message broker. The message broker plays a role in forwarding the message from publisher to subscriber and vice versa. Afterward, a new thread (Algorithm 1) is prepared to periodically check if there is any new topic at message broker.

Every time a new topic emerges, a new thread will be created with that new topic used as an ID. Each thread contains VRP solver procedure (Algorithm 2) that acts as a solution/route finder. Accordingly, a thread pool is provided to accommodate all threads from each topic arranged in arrival order. Since the topic is used as a thread ID, the thread pool size is equal to the number of census/survey location.

Each thread in the thread pool will be run consecutively, one session for each thread. Every running thread will find the solution by taking all M vehicles and $(unassigned)N$ locations into account. This solution finding is performed using VRP solver procedure which is explained in more detail in subsection III-B.

Algorithm 1 TopicWatcher

```
Input: $C$
Output: $T$
for $i = 1$ to $len(C)$ do
    for $j = 1$ to $N$ do
        if $(C_i == E_j)$ then
            $T_j = Thread(C_i, V_1...V_M, (unassigned)E_1...E_N)$
        end if
    end for
end for
return $T$
```

The VRP solver procedure in each thread will produce at least 1 route and at most M routes. Every route R_i that is produced will be published with a topic C_i. The message broker will inform the publisher the number of subscribers that receive the message. The subscribers' identities will remain unknown to the publisher.

If a route R_i has more than one message receiver, the thread with ID C_{R_i} will be cancelled. This annulment is aimed to prevent a solution/route that has been received by subscribers from being recomputed. There could be a condition when the running VRP solver with ID C_i does not produce the solution for topic C_i. In this situation, the topic C_i will be requeued in thread pool by only involving the subscriber as the enumerator. As a result, there is a guarantee that every topic C_i will have a solution/route.

Algorithm 2 VRPWorker

```
Input: $T$
Output: $R$
for $i = 1$ to $len(T)$ do
    $R$ = VRPSolver($T_i$)
    for $j = 1$ to $len(R)$ do
        $r$ = publish($C_{R_j}$, $R_j$)
        if $(r > 0)$ then
            cancelSolver($T_j$)
        else if $(C_{T_i} \notin C_{R_j})$ then
            $T_i = Thread(C_{T_i}, V_m, (unassigned)E_1...E_N)$
        end if
    end for
end for
return $R$
```

Despite the flexibility, the loose coupling characteristics also brings a shortcoming to publish/subscribe mechanism. The publisher's unawareness towards subscribers' identities caused the subscribers' location cannot be identified. Hence, a mechanism that can support information exchange between publisher and subscribers is needed. The idea is to use a shared memory contains subscribers' current location data that can be accessed by the publisher.

The whole processes described above, from detecting a new topic to obtaining the solution, will keep repeating until all customers are already assigned to each vehicle. In terms of data collection, the entire iterative processes will stop once all census/survey locations have been assigned to each enumerator. Figure 4 illustrates the workflow of the algorithm used in recommendation publisher.

B. VRP Solver

VRP solver is a module used for finding the best route/solution to visit all customers (census/survey locations) in a VRP/MDVRP. VRP solver is called in each thread in the thread pool (Algorithm 2). There are several algorithms that can be adopted to find the solution such as tabu search [6], adaptive large neighborhood search [7], fuzzy logic guided genetic algorithm [8], paralel iterated tabu search [9], hybrid algorithm combining iterated local search and set partitioning [10], hybrid genetic algorithm with adaptive diversity control [11], hybrid granular tabu search [12], and cooperative coevolution algorithms (CoEAs) [13]. In this research, VRP solver utilizes CoEAs since CoEAs generates a competitive *mean solution values* with relatively low CPU time compared to other algorithms [13].

The VRP solver works as follows:

1) Problem creation
   Create an instance of MDVRP that consists of M vehicles (enumerators) and N customers (locations).
2) Problem decomposition
   Decompose the MDVRP into several subproblems using

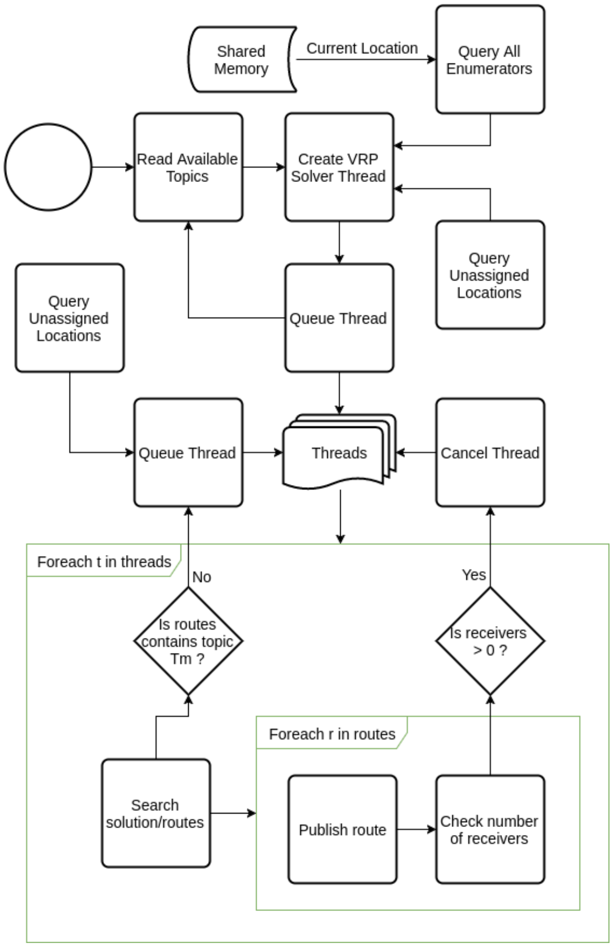

Fig. 4. Publisher Workflow

Nearest Insertion Heuristic (NIH) and Semi-greedy algorithm. Both of these algorithms are parts of CoEAs.

3) Individual evolution.
Before each evolution, each individual in every subproblem is evaluated to find the current best individual (CBI) of all subproblems. Then, evolve each individual to generate new individuals. Evaluates these new individuals to find the new best individual (NBI). If NBI is better than CBI, update CBI with NBI value. Otherwise, leave the CBI as it is. Repeat the evolution process until it reaches the time limit of 60 seconds or 40 seconds if there is no change happens to the CBI.

C. Message Broker

A message broker is a component responsible for routing the message from publisher to subscriber based on the subscribed topic [14]. A publish/subscribe system can have a single broker or multi-broker. In single broker architecture, all subscribers and publishers are connected to one single broker, while in multi-broker architecture, every subscriber or publisher can connect to any nearest broker. This multi-broker architecture is also called distributed publish/subscribe system [5] as depicted in Figure 5. In this research, the proposed design will implement distributed architecture since the census/survey locations are geographically scattered.

Algorithm 3 VRPSolver

```
Input: $T_i$
Output: $R$
P ← createProblem($V_1...V_{T_i}$, $E_1...E_{T_i}$)
*Decompose problem into S subproblem*
SP ← decomposeProblem(P)
while stopCriteriaUnmet do
  for $i = 1$ to $len(SP)$ do
    I ← $SP_i \rightarrow individuals$
    for $j$ to $len(I)$ do
      $O_j \leftarrow I_j \rightarrow evolve()$
    end for
    shrink($I$, $O$)
  end for
  updateBestIndividuals()
end while
$B \leftarrow getBestIndividuals()$
$R \leftarrow convert(B)$
return $R$
```

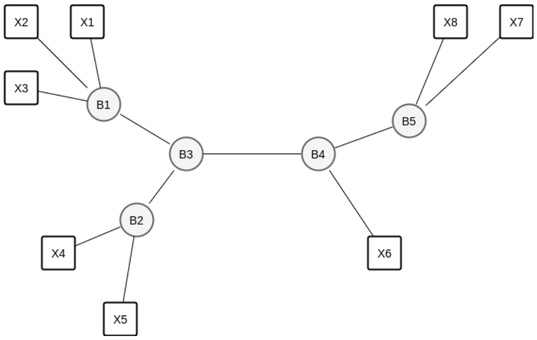

Fig. 5. The architecture of distributed publish-subscribe

IV. Result

The performance of the proposed system is assessed through a number of experiments. This experiment uses 2 (two) input data: Cordeau's data and real administrative data. The service time for these two data is randomly generated by following the normal distribution.

1) The Cordeau's data [6], which consists of 50 customers (locations) and 4 vehicles (enumerators). The distances between locations are computed using euclidean formula.
2) The real administrative data, which consists of 182 customers (locations) and 15 vehicles (enumerators). The distances and the time needed to travel between locations are measured using Google Maps Direction API [15].

These two data are tested using the proposed program, which combines Publish/Subscribe mechanism and CoEAs. The scenario used in testing is shown in the flowchart in Figure 6. The output of this test is compared against the output from the benchmark program that uses CoEAs algorithms only.

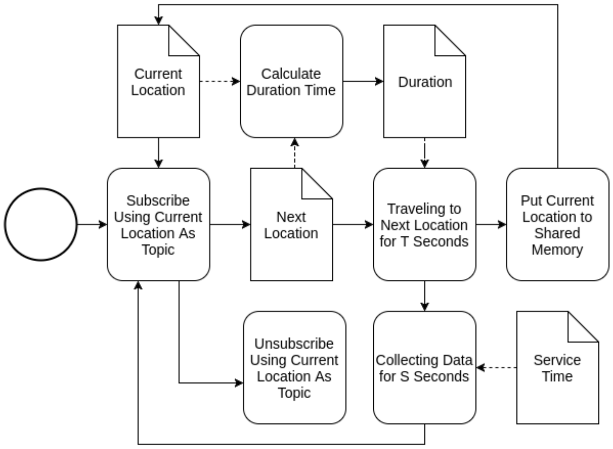
Current Location
Calculate Duration Time
Duration
Subscribe Using Current Location As Topic
Next Location
Traveling to Next Location for T Seconds
Put Current Location to Shared Memory
Unsubscribe Using Current Location As Topic
Collecting Data for S Seconds
Service Time

Fig. 6. The flowchart of experiment scenario

Both programs output the routes for each vehicle which can be illustrated as follows:

- *Vehicle* A = Loc1 → Loc5 → Loc15 → Loc6
- *Vehicle* B = Loc6 → Loc2 → Loc16 → Loc3
- *Vehicle* C = Loc4 → Loc8 → Loc14 → Loc 7
- *Vehicle* D = Loc9 → Loc10 → Loc11 → Loc12

where Loc is the location to visit.

The total time for each route is calculated by adding up the service time and the transport time of all locations in that route. Standard deviation is used to compare the accuracy of output from proposed program and output from benchmark program. Standard deviation is chosen as the metric because it gives the insight about how diverse the output is, whether the outputs are close to the average or spread out over a wide range. The idea is the smaller variance of total time between each enumerator, the more equal and fairer the enumeration. Therefore, a better program will have smaller standard deviation.

The test using Cordeau's data in normal condition shows that the pre-calculated routes generated by benchmark program yield a smaller total time (1,165,513.28 seconds) compared with that of the proposed program (1,165,706.84 seconds). However, the benchmark program has, a bigger standard deviation of 98,268.51 seconds, in contrast with the proposed program of 12997.91 seconds (Table I).

Similarly, the test using real administrative data in normal condition gives the result that pre-calculated routes from benchmark program return a smaller total time, yet a higher standard deviation compared with that of the proposed program. The benchmark program outputs a total time of 4,448,989.67 seconds and standard deviation of 119,720.84 seconds, while the proposed program gives total time of 4,559,658.67 seconds and standard deviation of 34,472.12 seconds (Table II).

V. Conclusion and Future Works

This paper proposes the use of Cooperative Coevolutionary Algorithm (CoEAs) combined with Publish/Subscribe mechanism to solve the real-time location recommendation problem in field data collection. Based on the test results, it is concluded that:

1) CoEAs algorithm outputs a more efficient route based on the total cost point of view, but,
2) The CoEAs algorithm cannot be directly used in census/survey location recommendation since it contains a huge standard deviation of the total cost for each route, thus,
3) The use of a real-time mechanism such as Publish/Subscribe mechanism is needed to overcome the inequality problem between each route. It means the tasks and locations will be more evenly distributed and every enumerator has equal workloads.

Publish/Subscribe mechanism is not the only option that can be used to develop a real-time system. Further studies using other mechanisms such as Push/Pull mechanism and Request/Reply mechanism is required for comparison. A research on analyzing the performance of each mechanism in varying network conditions are also necessary to create a simulation of real field conditions.

TABLE I
The comparison between CoEAs algorithm and Pub/Sub + CoEAs algorithm for the Cordeau data

Units	CoEAs	CoEAs + Pub/Sub
Total time (s)	1165513.28	1165706.84
Mean of time (s)	291378.32	291426.71
Std. Dev of time (s)	98268.51	12997.91

TABLE II
The comparison between CoEAs algorithm and Pub/Sub + CoEAs algorithm for the real field data

Units	CoEAs	CoEAs + Pub/Sub
Total time (s)	4448989.67	4559658.67
Mean of time (s)	296599.31	303977.24
Std. Dev of time (s)	119720.84	34472.12

References

[1] J. R. Montoya-Torres, J. Lpez Franco, S. Nieto Isaza, H. Felizzola Jimnez, and N. Herazo-Padilla, "A literature review on the vehicle routing problem with multiple depots," *Computers & Industrial Engineering*, vol. 79, pp. 115–129, Jan. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S036083521400360X

[2] J. Renaud, G. Laporte, and F. F. Boctor, "A tabu search heuristic for the multi-depot vehicle routing problem," *Computers & Operations Research*, vol. 23, no. 3, pp. 229–235, Mar. 1996. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0305054895O0026P

[3] A. P. Engelbrecht, "Coevolution," in *Computational Intelligence*. John Wiley & Sons, Ltd, 2007, pp. 275–283, dOI: 10.1002/9780470512517.ch15. [Online]. Available: http://onlinelibrary.wiley.com.proxy.lib.odu.edu/doi/10.1002/9780470512517.ch15/summary

[4] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, "The many faces of publish/subscribe," *ACM computing surveys (CSUR)*, vol. 35, no. 2, pp. 114–131, 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=857078

[5] G. Mhl, "Large-scale content-based publish-subscribe systems," Ph.D. dissertation, TU Darmstadt, 2002. [Online]. Available: http://tuprints.ulb.tu-darmstadt.de/274

[6] J.-F. Cordeau, M. Gendreau, and G. Laporte, "A tabu search heuristic for periodic and multi-depot vehicle routing problems," *Networks*, vol. 30, no. 2, pp. 105–119, Sep. 1997. [Online]. Available: http://onlinelibrary.wiley.com.proxy.lib.odu.edu/doi/10.1002/(SICI)1097-0037(199709)30:2⟨105::AID-NET5⟩3.0.CO;2-G/abstract

[7] D. Pisinger and S. Ropke, "A general heuristic for vehicle routing problems," *Computers & Operations Research*, vol. 34, no. 8, pp. 2403–2435, Aug. 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305054805003023

[8] H. C. W. Lau, T. M. Chan, W. T. Tsui, and W. K. Pang, "Application of Genetic Algorithms to Solve the Multidepot Vehicle Routing Problem," *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 2, pp. 383–392, Apr. 2010.

[9] J.-F. Cordeau and M. Maischberger, "A parallel iterated tabu search heuristic for vehicle routing problems," *Computers & Operations Research*, vol. 39, no. 9, pp. 2033–2050, Sep. 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305054811002826

[10] A. Subramanian, E. Uchoa, and L. S. Ochi, "A hybrid algorithm for a class of vehicle routing problems," *Computers & Operations Research*, vol. 40, no. 10, pp. 2519–2531, Oct. 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S030505481300021X

[11] T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins, "Implicit depot assignments and rotations in vehicle routing heuristics," *European Journal of Operational Research*, vol. 237, no. 1, pp. 15–28, Aug. 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S037722171301028X

[12] J. W. Escobar, R. Linfati, P. Toth, and M. G. Baldoquin, "A hybrid Granular Tabu Search algorithm for the Multi-Depot Vehicle Routing Problem," *Journal of Heuristics*, vol. 20, no. 5, pp. 483–509, Oct. 2014. [Online]. Available: http://link.springer.com.proxy.lib.odu.edu/article/10.1007/s10732-014-9247-0

[13] F. B. de Oliveira, R. Enayatifar, H. J. Sadaei, F. G. Guimares, and J.-Y. Potvin, "A cooperative coevolutionary algorithm for the Multi-Depot Vehicle Routing Problem," *Expert Systems with Applications*, vol. 43, pp. 117–130, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417415005771

[14] G. Banavar, T. Chandra, B. Mukherjee, J. Nagarajarao, R. E. Strom, and D. C. Sturman, "An efficient multicast protocol for content-based publish-subscribe systems," in *Distributed Computing Systems, 1999. Proceedings. 19th IEEE International Conference on*. IEEE, 1999, pp. 262–272. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=776528

[15] G. Google, "Google Maps Directions API," 2016. [Online]. Available: https://developers.google.com/maps/documentation/directions/