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 similiar workflow with Multi Depot Vehicle Routing Problem (MDVRP). The use of MDVRP to generate pre-calculated routes resulted in a total route costs with high standard deviation. 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 the routes produced by publish/subscribe paradigm combined with CoEAs are more prevalent in total route times among enumerators compared with pre-calculated routes produced by only CoEAs.**

I. Introduction

Data collection is one of the main duties of the statistical agency in every country. There are two data collection methods that are generally used: primary data collection and secondary data collection. In the primary data collection, the enumerators directly meet and interview the respondents, while in the secondary data collection, the statistical institution only compiles the data collected by other agencies. The primary data collection method requires the enumerators to be allocated to several census/survey areas that have been specified before. The allocation is usually made by assigning an equal number of locations on each enumerator.

The workflow of the data collection performs as follows:

1) The enumeration will start on a particular point. It could be the statistical office or the enumerator's house.
2) The enumerator moves to the first census/survey location and visits all respondents in that location.
3) 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), or more specifically, the Multi Depot Vehicle Routing Problem (MDVRP).

However, although the data collection shares similar workflow with MDVRP, the rough implementation of the MDVRP algorithm is not feasible to solve the allocation problems. This is because one important variables needed in the algorithm, the service time, is not available until after the enumerator finish visiting the census/survey location. Therefore, this research aims to propose an integration of MDVRP with pub/sub mechanism to create a location recommendation system for field data collection that can be used in real time.

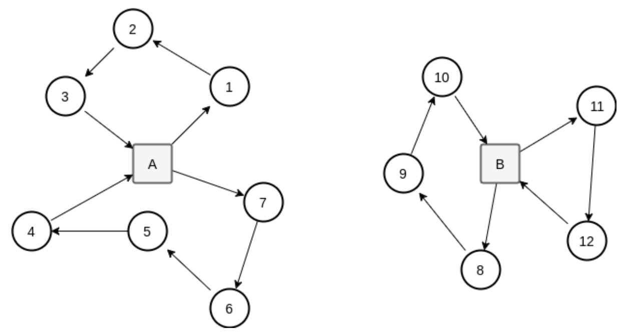

Fig. 1. An illustration for *Multi Depot* VRP

II. Related Works

A. Multi-Depot Vehicle Routing Problem

Multi-Depot Vehicle Routing Problem (MDVRP) is one of classic Vehicle Routing Problem (VRP) variance, where there are more than one depot that can be used [1]. Figure 1 shows an example of the solution for VRP problem that uses two depot and two vehicle routes which associated with each depot. Basically, a solution to this problem is a set of vehicle routes such that: (i) each vehicle route starts and ends at the same depot, (ii) each customer is served exactly once by one vehicle, (iii) the total demand on each route does not exceed vehicle 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. Let $G = (V, A)$ be a complete graph, where V is the set of nodes and A is the set of arcs. The nodes are partitioned into two subsets: the customers to be served, V_C = $\{1,..., N\}$, and the multiple depots V_D = $\{N+1,..., N+M\}$, with $V_C \cup V_D$ = V and $V_C \cap V_D$ = $\oslash$. There is a non-negative cost c_{ij} associated with each arc $(i, j) \in A$. The demand of each customer is d_i (there is no demand at the depot nodes). There is also a fleet of K identical vehicles, each with capacity Q. The service time at each customer i is t_i while the maximum route duration time is set to T. 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 the same as the time and distance units, so $w_{ij} = 1$.

In the mathematical formulation [3], binary variables x_{ijk} are equal to 1 when vehicle k visits node j immediately after node i. Auxiliary variables y_i are also used in the subtour elimination constraints.

Minimize

$$\sum_{i=1}^{N+M} \sum_{j=1}^{N+M} \sum_{k=1}^{K} c_{ij} x_{ijk}; \tag{1}$$

Subject to:

$$\sum_{i=1}^{N+M} \sum_{k=1}^{K} x_{ijk} = 1 (j = 1, ..., N); \tag{2}$$

$$\sum_{j=1}^{N+M} \sum_{k=1}^{K} x_{ijk} = 1 (j = 1, ..., N); \tag{3}$$

$$\sum_{i=1}^{N+M} x_{ihk} - \sum_{j=1}^{N+M} x_{hjk} = 0 \tag{4}$$

$$(k = 1, ..., K; h = 1, ..., N + M);$$

$$\sum_{i=1}^{N+M} \sum_{j=1}^{N+M} d_i x_{ijk} \leq Q (k = 1, ..., K); \tag{5}$$

$$\sum_{i=1}^{N+M} \sum_{j=1}^{N+M} (c_{ij} w_{ij} + t_i) x_{ijk} \leq T (k = 1, ..., K); \tag{6}$$

$$\sum_{i=N+1}^{N+M} \sum_{j=1}^{N} x_{ijk} \leq 1 (k = 1, ..., K); \tag{7}$$

$$\sum_{j=N+1}^{N+M} \sum_{i=1}^{N} x_{ijk} \leq 1 (k = 1, ..., K); \tag{8}$$

$$y_i - y_j + (M + N) x_{ijk} \leq N + M - 1; \tag{9}$$

$$for 1 \leq i \neq j \leq N and 1 \leq k \leq K;$$

$$x_{ijk} \in \{0, 1\} \forall i, j, k; \tag{10}$$

$$y_i \in \{0, 1\} \forall i; \tag{11}$$

The objective 1 minimizes the total cost. Constraints 2 and 3 guarantee that each customer is served by exactly one vehicle. Flow conservation is guaranteed through constraint 4. Vehicle capacity and route duration constraints are found in 5 and 6, respectively. Constraints 7 and 8 check vehicle availability. Subtour elimination constraints are in 9. Finally, 10 and 11 define x and y as binary variables.

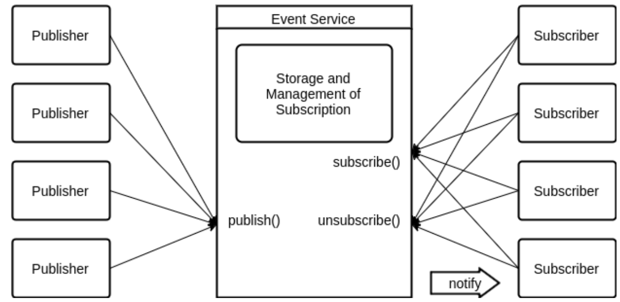

Fig. 2. Basic architecture in Pub/Sub [5]

B. Evolution Algorithms

Evolution Algorithms (EAs) is one of the optimization algorithm families that are inspired by nature. In the evolution algorithm, there are selection and reproduction processes to purify the population from the solution candidates. According to Engelbreacht [4], in general, the EAs lifecycle begins with randomly configuring the initial population. Afterwards, for each iteration, every individual in the population is evaluated using an objective function. Then, each solution candidate will be labeled with a fitness value based on their evaluation result. These solution candidates will then commit a reproduction and create the next generation. The whole process will be repeated using the new-generation individuals. These iterations will stop once it hits a certain time limit, which is 60 seconds.

Coevolutionary algorithms (CoEAs) is a development of EAs. In standard EAs, evolution is usually viewed as the population attempts to adapt in a fixed physical environment. In contrast, coevolutionary algorithms (CoEAs) realize that in natural evolution the physical environment is influenced by other independently-acting biological populations [4]. Based on the interaction of each species, CoEAs can be distinguished into two categories: competitive and cooperative [4]. In the competitive coevolution, each individual competees with other individuals in the same group. Meanwhile, in cooperative coevolution, the interaction between species is mutualistic, or at least, not harming each other.

C. Publish/Subscribe Paradigm

Publish/subscribe interaction is one of the communication methods between client and server. The interaction paradigm in publish/subscribe is the presence of subscribers that have an interest in a certain event or the pattern of an event. These subscribers will be notified by the publisher about the event they are interested in [5].

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, who acts as the events producer, and the subscriber who acts as the event consumer.

One of the benefits of publish/subscribe mechanism is the loose coupling characteristic [5] between the publisher and the subscriber. The separation of information (decoupling)

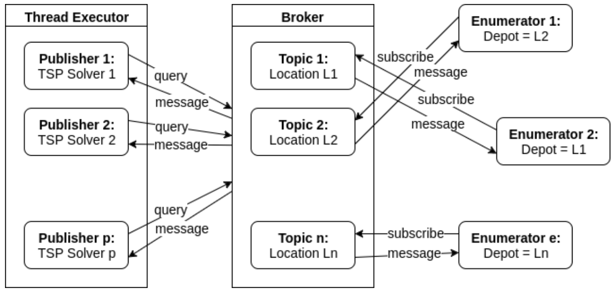

Fig. 3. The system outline

between publisher and subscriber can be divided into 3 (three) dimension, which are:

1) Space decoupling.
   In doing the interaction, both publisher and subscriber do not need to know each other. The publisher sends an event and subscriber indirectly receive the event through the event service. The publisher does not hold any reference regarding 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 even though he is in a disconnected state. Similarly, the subscriber can still receive the event, albeit disconnected.
3) Synchronizing decoupling.
   Publisher will not be blocked when producing the event. 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 are not interacting synchronously.

III. Proposed Solution

The drawback of using MDVRP for location recommendation in field data collection is the absence of time service data, which is very important in the recommendation calculation. Thus, MDVRP needs to be integrated with the real-time mechanism. There are several mechanisms that can be adopted in a real-time system such as Web Service, Remote Procedure Call (RPC), message passing, and publish/service [5].

The publish/subscribe mechanism is used in this research because it has the *loose coupling* characteristic. Furthermore, publish/subscribe mechanism is also suitable for an information-driven system [6]. Because of this characteristic, the publish/subscribe mechanism can work asynchronously, where request and reply do not have to be processed in sequence. The publisher and subscriber can conduct an offline publish and offline subscribe, respectively.

Figure 3 shows the system outline that consists of 3 (three) main components: the publisher, the subscriber, and the message broker. The communication between publisher and subscriber is built upon a similarity in either event or topic. In this research, the topic that is used as the basis for the communication between publisher and subscriber is the *current location* of each subscriber.

A. Recommendation Publisher

VRP solver is implemented on the publisher's side, who will search for solution/route and then publish it to the subscriber via the message broker. The subscriber's current location is used as a topic. Afterward, a thread (Algorithm 1) is prepared that will periodically check if there is any new topic at message broker.

Each time a new topic emerge, a new thread will be created and that new topic will be used as an ID for that thread. Each thread contains VRP solver procedure (Algorithm 2) that will do the solution/route finding. Accordingly, a threadpool is provided to contain all threads from each topic in order of arrival time. Since topic is used as a thread ID, the threadpool size is equal to the number of census/survey location.

Each thread in thread pool will be run in order, one session for each thread. Every running thread, with a topic as an ID, will find the solution by embroiling all M vehicles and $(unassigned)N$ locations. The solution finding is done 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 process in 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 topic C_i. After being published, the message broker will inform the publisher about the number of subscribers that receive the message. The identity of the subscribers that receive the message 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 is a condition when the running VRP solver with ID C_i does not produce the solution for topic C_i. In this kind of situation, the topic C_i will be requeued with only including the subscriber as the enumerator. By doing this, there is a guarantee that the topic C_i will obtain a solution/route.

As mentioned before, the advantage of using publish/subscribe mechanism is its loose coupling characteristic that makes it flexible. But this benefit could also be deemed as a shortcoming. The publisher's unawareness towards subscribers' identities has caused the current location of all subscribers cannot be identified. Hence, we need a mechanism that can support information exchanging between publisher

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

and subscribers. The idea is to use a shared memory contains the current location data of all subscribers. By accessing this shared memory, the publisher will be able to determine the subscribers' current locations.

The process above will keep repeating until all enumeration locations have been assigned to the subscribers (enumerators). Figure 4 illustrates the workflow of the algorithm used in recommendation publisher.

B. VRP Solver

VRP solver is a module used for finding a solution or route that will be visited. VRP solver is called in each thread in the thread pool (Algorithm 2). There are several algorithms that can be used in finding a solution in a MDVRP problem such as tabu search [7], adaptive large neighborhood search [8], fuzzy logic guided genetic algorithm [9], paralel iterated tabu search [10], hybrid algorithm combining iterated local search and set partitioning [11], hybrid genetic algorithm with adaptive diversity control [12], hybrid granular tabu search [13], and cooperative coevolution algorithms (CoEAs) [14]. In this research, we will use coevolutionary algorithms (CoEAs) since CoEAs generates a competitive *mean solution values* with relatively low CPU time compared to other algorithms [14].

The steps that are used in VRP solver that adopts CoEAs algorithm are as follows:

1) Create problem
MDVRP problem that is designed consists of a number of vehicles V and edges E that are assigned to each thread.
2) Decompose problem
MDVRP problem is decomposed into several subproblems.
3) Evolve.
Each individual in every subproblem will be evolved. At the end of every evolution, new indivuals will be evaluated. If the new generated individuals are better than the previous ones, the solution is accepted and the best individuals will be updated. The evolution will keep happening until it reach the time limit of 60 seconds, or 40 seconds without any changes on the best individuals.

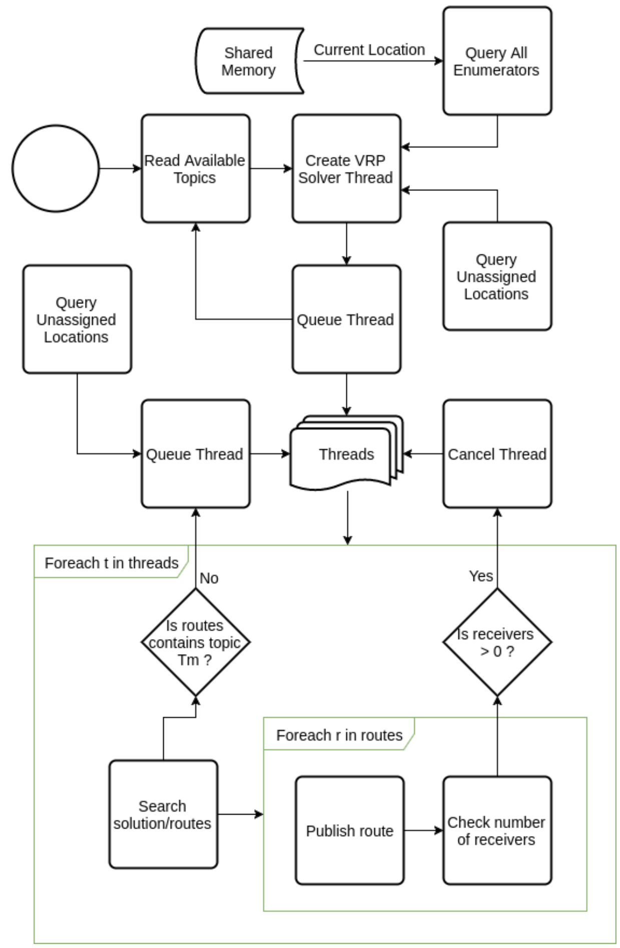

Fig. 4. Publisher Workflow

Algorithm 3 VRPSolver

Input: T_i
Output: R

P $\leftarrow$ createProblem($V_1...V_{T_i}$, $E_1...E_{T_i}$)
Decompose problem into S subproblem
SP $\leftarrow$ decomposeProblem(P)
while stopCriteriaUnmet **do**
  for $i = 1$ to $len(SP)$ **do**
    I $\leftarrow SP_i \rightarrow individuals$
    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

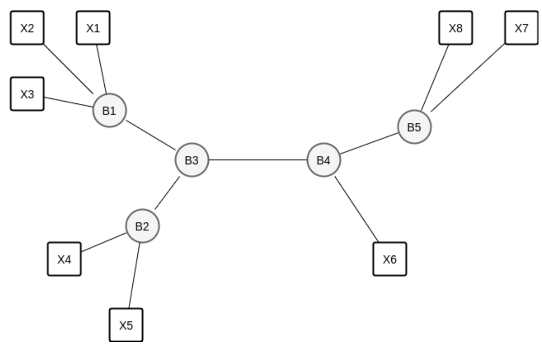

Fig. 5. The architecture of distributed publish-subscribe

C. Message Broker

A message broker is a component responsible for distributing the message routing from publisher to subscriber based on the topic that is subscribed [15]. 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 the nearest broker. This multi-broker architecture is also called distributed pub/sub system [6] as depicted in Figure 5. In this research, the proposed design will implement distributed architecture since the census/survey locations are geographically scattered.

IV. Testing

The performance of the proposed system was assessed through a number of experiments. This experiment uses 2 (two) types of data: Cordeau and field data. The service time for these two data types will be randomly generated by following the normal distribution.

1) Data from Cordeau [7], consists of 50 customers and 4 vehicles. The distance between locations are computed using euclidean formula.
2) Data from real field condition, which is consists of 182 customers and 15 vehicles. The distance and the time needed to travel between locations are measured using Google Maps Direction API [16].

These two data are tested using the proposed program, which is based on Pub/Sub and CoEAs. The output from this test is compared with another output from the benchmark program that uses CoEAs algorithms without Pub/Sub mechanism. From both program, we will obtain outputs in the form of the routes for each vehicle that can be illustrated as follows:

- *Vehicle* A = Loc1 → Loc5 → Loc15 → Loc6
- *Vehicle* B = Loc 6 → Loc2 → Loc16 → Loc3
- *Vehicle* C = Loc4 → Loc8 → Loc14 → Loc 7
- *Vehicle* D = Loc9 → Loc10 → Loc11 → Loc12

The total cost for each route is calculated by adding up the service time and the travel time from all locations in that route. The standard deviation of all routes in each program (the proposed program and the benchmark program) is used as a metric for comparing the outputs. A better program will result in a smaller standard deviation. Standard deviation is chosen as the metric because it represents the real field condition. If the time variance between each enumerator is smaller, the enumeration completion will be equal and fair.

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

From the test using Cordeau data in normal condition, we obtain a result that the pre-calculated routes generated by CoEAs algorithm yield a smaller total time of 1,165,513.28 seconds compared with the result from Pub/Sub + CoEAs of 1,165,706.84 seconds. However, the standard deviation of the output from the CoEAs algorithm is bigger than that of Pub/Sub + CoEAs algorithm, which are 98,268.51 seconds and 12997.91 seconds (Table I), respectively. Similarly, the test using real field data in normal condition give the result that pre-calculated routes using CoEAs algorithm yield a smaller total time, yet a higher standard deviation compared with that of Pub/Sub + CoEAs. The CoEAs algorithm yield a total time of 4,448,989.67 seconds and standard deviation of 119720.84 seconds, while the Pub/Sub + CoEAs algorithm gives a total of 4,559,658.67 seconds and standard deviation of 34472.12 seconds (Table II).

V. Conclusion and Future Works

This paper proposes the use of Cooperative Coevolutionary Algorithm (CoES) that is 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 from total cost standpoint, but,
2) The CoEAs algorithm cannot be directly used in census/survey location recommendation because there is a huge standard deviation of the total cost for each route, thus,
3) The use of a real-time mechanism such as Pub/Sub mechanism is feasible to overcome the inequality problem between each route. It means the tasks and locations are

more evenly distributed and every enumerator has equal workloads.

Pub/Sub mechanism is not the only mechanism that can be used as a real-time mechanism. Further studies using other mechanisms, such as: Push/Pull mechanism, Req/Rep mechanism is needed as a comparison. Research on how the performance of each of these mechanisms in varying network conditions are also required to simulate actual field conditions.

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] R. V. Kulkarni and P. R. Bhave, "Integer programming formulations of vehicle routing problems," *European Journal of Operational Research*, vol. 20, no. 1, pp. 58–67, Apr. 1985. [Online]. Available: http://www.sciencedirect.com/science/article/pii/037722178590284X

[4] 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

[5] 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

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

[7] 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

[8] 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

[9] 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.

[10] 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

[11] 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

[12] 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

[13] 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

[14] 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

[15] 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

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