Real-Time Location Recommendation System 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 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 MDVRP based on 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 enumerator's total route times compared with the pre-calculated routes produced by MDVRP based on 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 to several census/survey areas. The allocation is usually conducted by assigning an equal number of locations to 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.

Although sharing similar workflow with MDVRP, the rough implementation of the MDVRP algorithm is not feasible for solving the allocation problems. Solutions or routes obtained using the MDVRP algorithm often result in uneven time of completion. This is because the MDVRP algorithm when used in the search solution uses the assumption that service time at each location is evenly distributed, whereas the facts vary greatly.

The inclusion of service time variables that match the field conditions in the search for a solution is not possible, because variable service time will not be available until after the enumerator finishes visiting the census/survey location. Therefore the MDVRP algorithm needs to be integrated with a communication mechanism to create a real-time location recommendation system for field data collection. This study aims to propose a system that integrates the MDVRP algorithm based on CoEAs and publish/subscribe mechanism.

II. Related Works

A. MDVRP based on Coevolution Algorithms (CoEAs)

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 associated 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.

Coevolutionary algorithms (CoEAs) is one of many algorithms [2]–[9] that can be used to solve MDVRP. CoEAs realize that in natural evolution the physical environment is influenced by other independently-acting biological populations [10]. Based on the interaction of each species, CoEAs can be distinguished into two categories: competitive and cooperative [10]. 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.

The use of CoEAs for solving MDVRP has been studied by several researchers, one of which is de Oliveira et al. [9]. De Oliveira's algorithm consists of 2 (two) stages: initiation

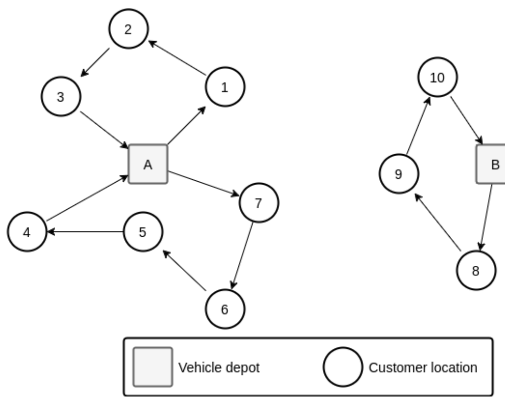

Fig. 1: An illustration for *Multi Depot* VRP

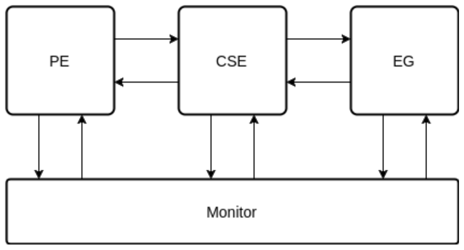

Fig. 2: Architecture of Parallel Modules [9]

and evolution. In initiation stage, the population is built using Nearest Insertion Heuristic (NIH) and NIH-based semi-greedy. Then, in the evolution stage, this population will be evolved using a parallel module, which consists of 3 submodules: Population Evolve (PE), Complete Solution Evaluation (CSE), and Elite Group (EG) submodules. This stage is controlled by a monitor module to ensure that all processes run well. Figure 2 shows the parallel architecture proposed by de Oliveira.

B. Publish/Subscribe Paradigm

The publish/subscribe interaction is a communication pattern between publisher (server) and subscribers (clients). The subscribers are the parties who have interests in a certain event/topic and will get a notification about the event/topic they are interested in [11]. One of the benefits of using publish/subscribe mechanism is the *loose coupling* characteristic [11] between the publisher and the subscriber.

The basic model of the publish/subscribe system relies on the event notification service that provides storage and management of subscription. This service acts as a mediator between the publisher (the event's producer) and the subscriber (the event's consumer).

III. Proposed Solution

Using pure MDVRP algorithm for location recommendation comes with one major drawback: the absence of service time data that match the field conditions which is very important for computing the recommendation. Thus, MDVRP algorithm needs to be integrated with a communication mechanism such as Web Service, Remote Procedure Call (RPC), message passing, and publish/subscribe mechanism [11].

In this research, the publish/subscribe mechanism is chosen because, in addition to its loose coupling characteristic [11], it is also suitable for an information-driven system [12]. *Loose coupling* allows the publish/subscribe mechanism to work asynchronously, meaning the request and the reply do not have to be processed in sequence. Publishing and subscribing activities can still be done even if one party is offline.

Figure 3 shows the outline for the proposed system that 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

Communication between publishers and subscribers uses a topic as a basis. This communication is done indirectly through a message broker. Unfortunately, the publish/subscribe mechanism makes it impossible for brokers to notify when new topics are subscribed. Therefore, it is necessary to check regularly by using a topic watcher to find out new topics that are subscribed. Algorithm 1 shows the referred algorithm.

Every time a new topic (subscribers current location) emerges, a new thread will be created using that new topic as an ID. Accordingly, a thread pool is provided to accommodate all threads in an arrival order. By the time all locations has been assigned to each enumerator, the thread pool size will be equal to the number of topics (census/survey locations).

Each thread will be running consecutively, one session for each thread. The reason why only one thread is executed is to avoid the recommended location conflict as illustrated in Figure 4. The current running thread calls a global VRP solver procedure that acts as a solution/route finder. VRP solver procedure takes into account all M enumerators and $(unassigned)N$ locations. This aims to prevent a local best solution, where 'next location' is the best location from one vehicle perspective only.

The VRP solver procedure in each thread will produce at least 1 route and at most M routes. Every generated route R_i

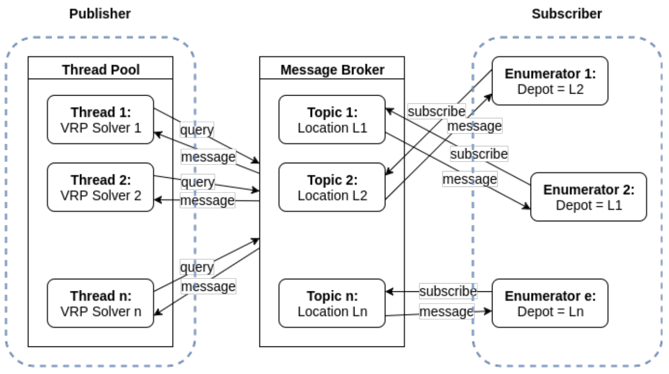

Fig. 3: The outline of the proposed system

Algorithm 1 Topic Watcher

```
Input: $None$
Output: $None$
    $TP$ = Threadpool
    $N$ = Number of locations
    $M$ = Number of enumerators
 1: while true do
 2:     $C \leftarrow readAvailableTopicFromBroker()$ // channel
 3:     for $m = 1$ to $len(C)$ do
 4:         for $n = 1$ to $N$ do
 5:             if ($C_m == L_n$) then
 6:                 $T_n = Thread(C_m, E_1...E_M, (unassigned)L_1...L_N)$
                    // E = enumerator
 7:                 submitThreadToThreadpool($T_n$, $TP$)
 8:             end if
 9:         end for
10:     end for
11: end while
```

Algorithm 2 Recommendation Publisher

```
Input: $TP$ // threadpool
Output: $None$
 1: while true do
 2:     $T$ = popFirstThreadOrWaitNewThreadFromThread-pool()
 3:     $R$ = VRPSolver($T$)
 4:     for $j = 1$ to $len(R)$ do
 5:         $r$ = publish($C_{R_j}$, $R_j$)
 6:         if ($r > 0$) then
 7:             cancelSolver($T_j$)
 8:         else if ($C_{T_i} \notin C_{R_j}$) then
 9:             $T_i = Thread(C_{T_i}, V_m, (unassigned)E_1...E_N)$
10:         end if
11:     end for
12: end while
```

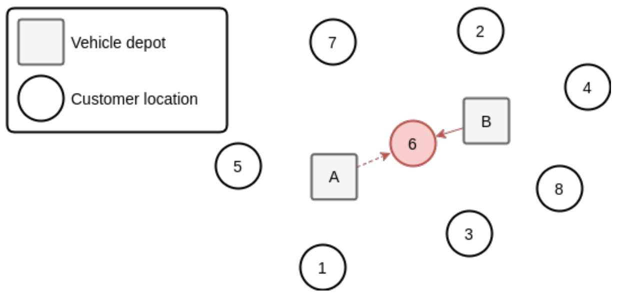

Fig. 4: The illustration of recommendation conflict

will be published and coupled with a topic C_i. The message broker will inform the publisher the number of subscribers receiving the message. If a route R_i has at least one subscriber, the thread with ID C_{R_i} will be canceled. This annulment is aimed to prevent a solution/route that has been received by a subscriber from being recomputed. The above steps are listed on Algorithm 2.

It could happen that the running VRP solver with ID C_i fails to produce the solution for topic C_i. In this situation, the topic C_i will be re-queued in the thread pool by involving only the subscriber of that particular topic. This mechanism guarantees that every topic C_i will have a solution/route.

Despite the flexibility, the loose coupling characteristic also brings a shortcoming to publish/subscribe mechanism. The publisher's unawareness towards subscribers' identities caused the subscribers' location cannot be identified. Hence, a technique 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 being repeated 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.

B. VRP Solver

VRP solver is a module used for finding the best route/solution to visit all customers (census/survey locations). VRP solver is implemented on the publisher's side and called in each thread in the thread pool. In this research, VRP solver utilizes CoEAs which generates a competitive *mean solution values* with relatively low CPU time compared to other algorithms [9].

The VRP solver works as follows:

1) Problem creation
   Create a MDVRP instance that consists of M vehicles (enumerators) and N customers (locations).
2) Problem decomposition
   Decompose the MDVRP into several subproblems using Nearest Insertion Heuristic (NIH) and Semi-greedy algorithm, which 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, each individual is evolved to generate new individuals. These new individuals will be evaluated 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 a particular 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 [13]. 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

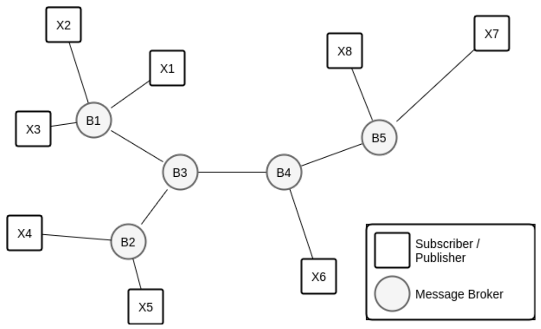

Fig. 5: The illustration of distributed publish-subscribe architecture

is also called distributed publish/subscribe system [12] as depicted in Figure 5. In this research, the proposed design will implement distributed architecture as the census/survey locations are geographically scattered.

IV. Result

A. Experimental Setup

Implementation. Each component in this proposed system is implemented differently. Recommendation publisher is implemented in `Python 2.7`. On the other hand, VRP Solver with the CoEAs algorithm is coded using `C++` and compiled with `C++ compiler 5.4.0 20160609`. Furthermore, the message broker is developed using `Redis Cluster 3.2.6` with 6 nodes: 3 nodes for masters and the other 3 nodes for the slaves.

Environment. The experiment is conducted on Elementary OS Loki 64bit operating system and Quad-Core Intel Core i3-4030U CPU @ 1.90GHz with 4 GB DDR3 RAM. Each Redis node is executed as a `Docker` container on Debian Jessie 64bit operating system and Quad-Core Intel Core i3-4030U CPU @ 1.90GHz with 4 GB RAM.

Dataset. This experiment is conducted using real administrative data, which consists of 182 locations and 15 enumerators. The distances and the time needed to travel between locations are measured using Google Maps Direction API [14]. While the service time for these two data is randomly generated based on Sudman [15] by following the normal distribution.

Scenario. The test will be run using 2 (two) scenarios: scenario of normal condition and delay condition. The dataset for testing in normal condition is generated by 4 (four) instances, where each instance has a different service time. While the dataset for testing in delay conditions is generated by 7 (seven) instances, where each instance has a different service time and also has varying delay.

Metric. The output from the proposed program (MDVRP algorithm based on CoEAs with publish/pubscribe mechanism) is compared against the output from the benchmark program that uses MDVRP algorithm based on CoEAs only. Both programs result in the routes for each vehicle which can be illustrated as follows:

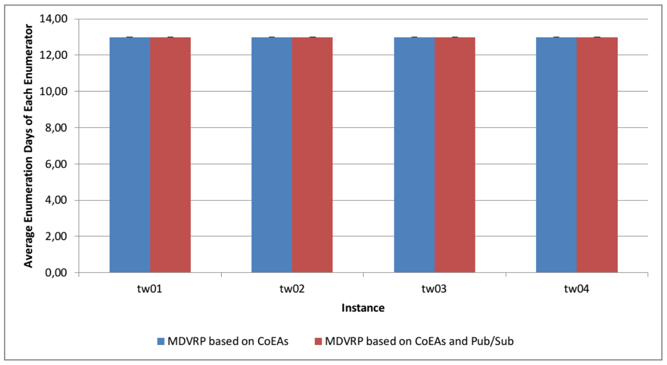

Fig. 6: The average days spent by each enumerator from 100 tests normal condition on field dataset

- *Enumerator* A = Loc1 $\rightarrow$ Loc5 $\rightarrow$ Loc15 $\rightarrow$ Loc6
- *Enumerator* B = Loc6 $\rightarrow$ Loc2 $\rightarrow$ Loc16 $\rightarrow$ Loc3
- *Enumerator* C = Loc4 $\rightarrow$ Loc8 $\rightarrow$ Loc14 $\rightarrow$ Loc 7
- *Enumerator* D = Loc9 $\rightarrow$ Loc10 $\rightarrow$ Loc11 $\rightarrow$ Loc12

where Loc is the location to visit.

The total days spent for each enumerator route is calculated by summing up the service time and the transport time of all locations in that route, which every day is limited by 8 (eight) work hours. After that, the total days and standard deviation from all enumerators can be calculated.

Standard deviation is chosen as a metric because it represents the actual condition, where the low standard deviation means the total number of days spent more evenly for all enumerators. Therefore, a better program will have a smaller standard deviation.

B. Experimental Result

Normal scenario. In the testing process, each dataset instance is executed 100 times to prove that the results obtained during the test have a low standard error. Based on the test result using field data and using the assumption of normal condition, where there are no obstacles in enumeration, all tested instances resulted that the average of enumeration day between the proposed system and the comparison program are the same, that is 13 days for each enumerator, as depicted in Figure 6. In terms of standard deviation, however, it shows that the proposed system yields a lower standard deviation on all instances, as depicted in Figure 7.

In line with the average number of enumeration days and standard deviation, using the proposed system, the amount of time spent waiting until all enumerators complete their enumeration is lower than the comparison program, as figured in Figure 8. So it can be concluded the proposed system is more efficient than the comparison system.

Based on the testing of normal conditions, it can be concluded that the proposed system can provide better recommendations. It means, the variation of the day of enumeration using the proposed system is lower, so the time required to wait for all enumerators to complete the enumeration is lower than the proposed program.

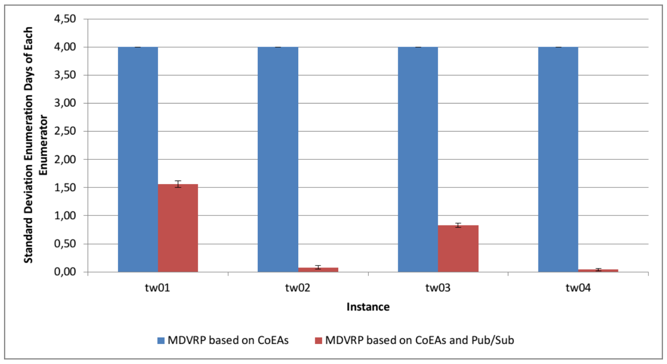

Fig. 7: The standard deviation days spent by each enumerator from 100 tests normal condition on field dataset

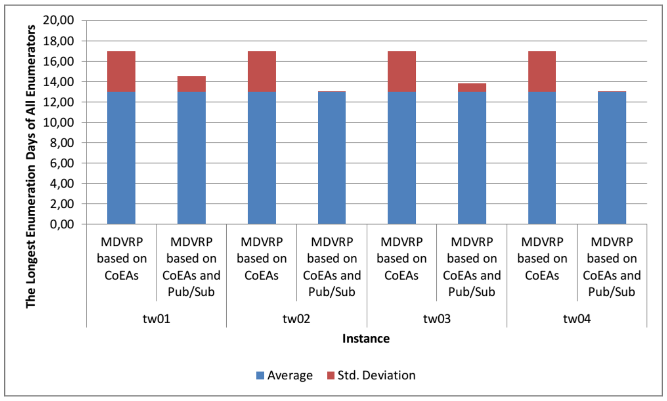

Fig. 8: The maximum days spent by each enumerator from 100 tests normal condition on field dataset

Delay scenario. Similar to testing in a normal scenario, testing on a delay scenario is also executed 100 times each instance . Contrast with the results of the tests in the normal scenario, the results obtained in the test with the delay scenario show that the average day spent by an enumerator in the proposed system is longer than the benchmark program in all instances, as figured in Figure 9. However, the standard deviation of the proposed system is lower than that of the benchmark program, as depicted in Figure 10, so the proposed system is more evenly distributed in term of day of the enumeration.

Although the average number of enumeration days using the proposed system takes longer time, the proposed system is still more efficient when applied to relatively low delay. Figure 11 shows that using the proposed system, the total number of days required to wait until all enumerators completed their enumeration is lower in the instance with the relatively low delay, which are d01, d02, and d04. While in the instance with high delay, which are d03, d05, and d06, the benchmark program gives more efficient results.

V. Conclusion and Future Works

This paper is aimed to solve the problem found in MDVRP based on CoEAS when used for generating a recommendation

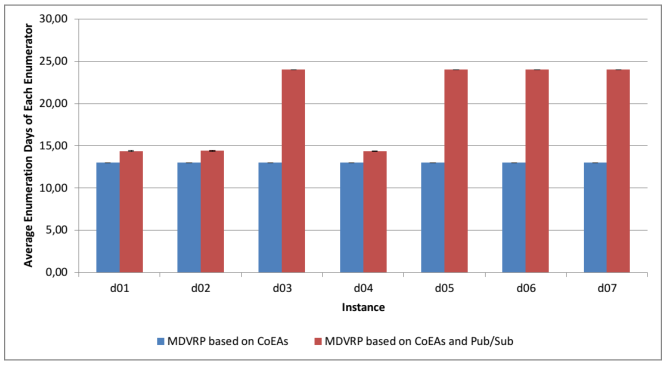

Fig. 9: The average days spent by each enumerator from 100 tests delay condition on field dataset

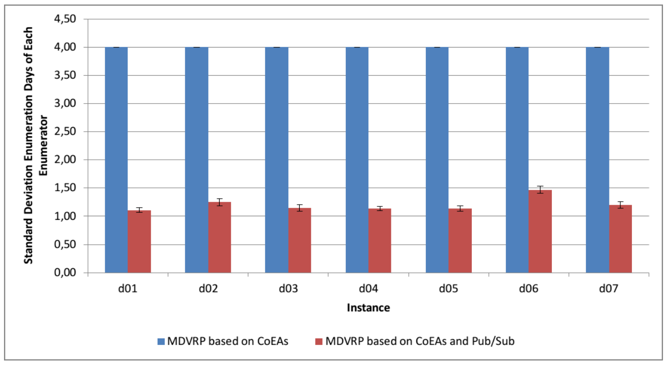

Fig. 10: The standard deviation days spent by each enumerator from 100 tests delay condition on field dataset

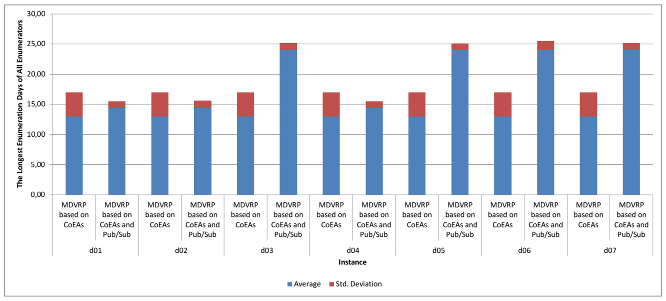

Fig. 11: The maximum days spent by each enumerator from 100 tests delay condition on field dataset

for enumeration locations. The proposed solution is to combine MDVRP based on CoEAS with publish/subscribe mechanism. The proposed system consists of 2 main components: the publisher and the message broker. The publisher contains a VRP solver which utilizes CoEAs algorithm.

Based on the results of the test using field data, the results show that the proposed system provides better results than the benchmark program on a scenario with no delay and a low delay scenario. While in scenarios with high delay, benchmark programs provide better results than proposed systems.

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 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.

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.-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

[3] 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

[4] 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.

[5] 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

[6] 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

[7] 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

[8] 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

[9] 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

[10] 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

[11] 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

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

[13] 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

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

[15] S. Sudman, "Time Allocation in Survey Interviewing and in Other Field Occupations," *The Public Opinion Quarterly*, vol. 29, no. 4, pp. 638–648, 1965. [Online]. Available: http://www.jstor.org.proxy.lib.odu.edu/stable/2747041