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 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. 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.

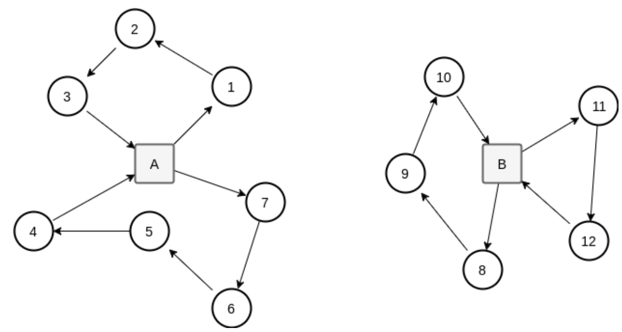


Fig. 1: An illustration for *Multi Depot* VRP

II. Related Works

A. 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 [6] [7] [8] [9] [10] [11] [12] [13] that can be used to solve MDVRP. CoEAs 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 coevolu-

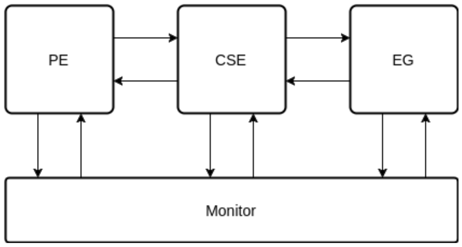


Fig. 2: Architecture of Parallel Modules [13]

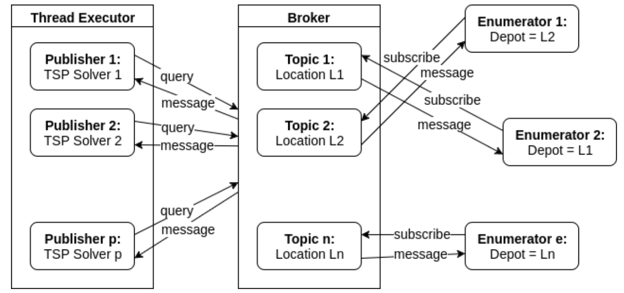


Fig. 3: The outline of the proposed system

tion, 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. [13]. De Oliveira's algorithm consists of 2 (two) stages: initiation 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 [4]. One of the benefits of using publish/subscribe mechanism is the *loose coupling* characteristic [4] 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 for location recommendation comes with one major drawback: 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]. *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 be done 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

The publisher uses subscriber's current location as a topic to find the best solution/route. This route will be published to the 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 regularly check if there is any new topic at message broker.

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 from each topic 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 current running thread calls a global VRP solver procedure (Algorithm 2) that acts as a solution/route finder. VRP solver procedure takes into account all M vehicles and $(unassigned)N$ locations, 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 generated route R_i will be published and coupled with a topic C_i. The message broker will inform the publisher the number of subscribers receiving the message. The subscribers' identities will remain unknown to the publisher.

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.

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 (vehicle) of that particular topic. This mechanism guarantees 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 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. 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). VRP solver is implemented on the publisher's side and called in each thread in the thread pool (Algorithm 2). In this research, VRP solver utilizes CoEAs which generates a competitive *mean solution values* with relatively low CPU time compared to other algorithms [13].

The VRP solver works as follows:

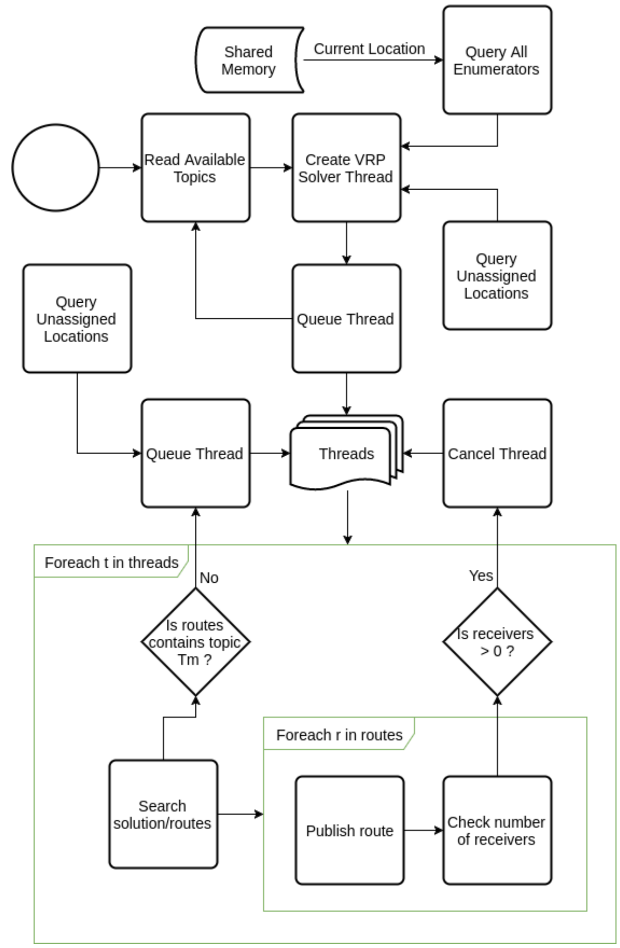


Fig. 4: The Publisher Workflow

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 [14]. A publish/subscribe system can have a single broker or multi-broker. In single broker architecture, all subscribers

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 → individuals
        for j to len(I) do
            O_j ← I_j → evolve()
        end for
        shrink(I, O)
    end for
    updateBestIndividuals()
end while
B ← getBestIndividuals()
R ← convert(B)
return R
```

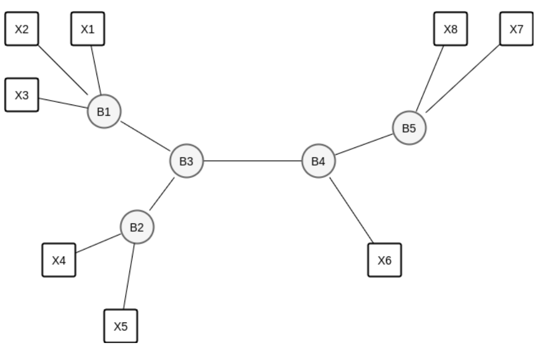


Fig. 5: The architecture of distributed publish-subscribe

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 as the census/survey locations are geographically scattered.

IV. Result

Each component in this proposed system is implemented differently. Recommendation publisher is written in `Java` and compiled with `Java Compiler 1.8.0_121`. 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.

The system testing 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

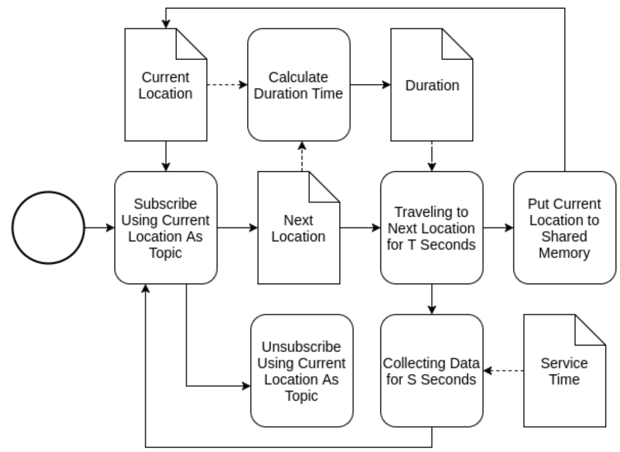


Fig. 6: The flowchart of client application

operating system and Quad-Core Intel Core i3-4030U CPU @ 1.90GHz with 4 GB RAM.

A client application is designed to simulate the real condition, which is performed as follows:

1) Client makes a subscription using current location as a topic.
2) Client waits until the server (publisher) provides a response that is the next-visited location.
3) Client travels to the next location.
4) Client conducts the enumeration.
5) Client stores its current location in shared memory.

Figure 6 represents the flowchart for client simulation.

This experiment is conducted using 2 (two) input data:

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 [16].

The service time for these two data is randomly generated based on Sudman [15] by following the normal distribution.

The output from the proposed program (Publish/Subscribe mechanism + MDVRP based on CoEAs) is compared against the output from the benchmark program that uses CoEAs algorithms only. Both programs result in 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 summing 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 both proposed and benchmark program. Standard deviation is chosen as the metric because it gives the insight

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

Units	CoEAs	CoEAs + Pub/Sub
Total time (s)	**905,018.39**	905,281.55
Mean of time (s)	**226,254.60**	226,320.39
Std. Dev of time (s)	49,715.99	**15,496.22**

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)	**3,373,390.19**	3,676,718.19
Mean of time (s)	**224,892.68**	245,114.55
Std. Dev of time (s)	91,123.23	**37,261.85**

about the output's diversity, whether it is 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 a smaller standard deviation.

(Figure 7a) and (Figure 7b) show the routes generated by the benchmark and proposed program, respectively, using Cordeau's data in normal condition. The test result, as can be seen in Table I, shows that the pre-calculated routes generated by benchmark program yield an insignificant 0.029% lower total time (905,018.39 seconds) compared with that of the proposed program (905,281.55 seconds). However, the benchmark program has a standard deviation of 49,715.99 seconds, which is more than 3 (three) times higher than that of the proposed program (15,496.22 seconds).

Similarly, the test using real administrative data in normal condition, as can be seen in Table II, shows 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 3,373,390.19 seconds and standard deviation of 91,123.23 seconds, while the proposed program gives a total time of 3,676,718.19 seconds and standard deviation of 37,261.85 seconds. The proposed program returns an output with a 0.52% higher total time, yet a 59.1% lower standard deviation, which is almost 3 (three) times lower than that of the benchmark program. (Figure 8a) and (Figure 8b) show the routes generated by the benchmark and proposed program, respectively, using real administrative data in normal condition.

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

The proposed system is tested using Cordeau's data and real administrative data, which shows that routes generated by proposed program are more prevalent in total route times compared with the pre-calculated routes produced by the benchmark program.

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. 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.

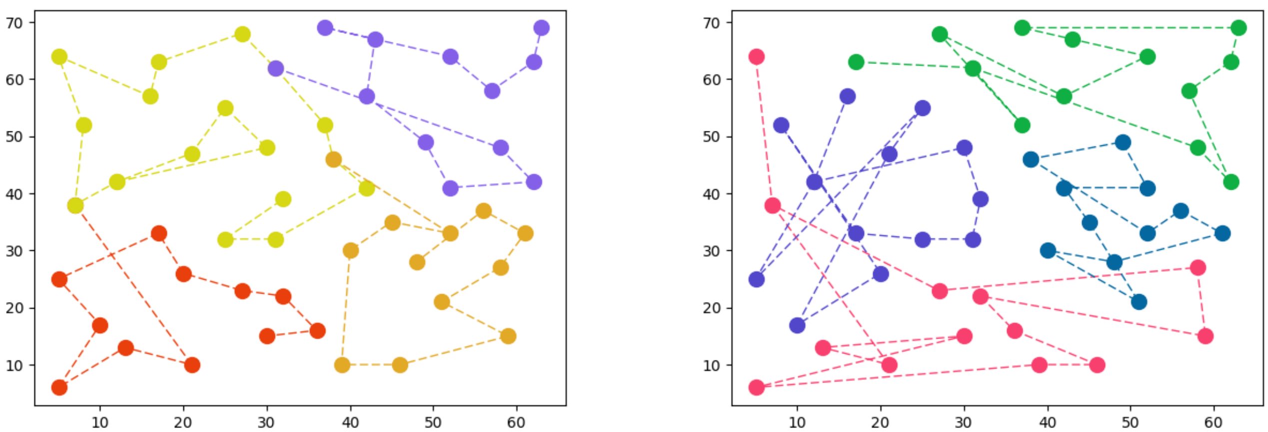
[9] J.-F. Cordeau and M. Maischberger, "A parallel iterated tabu search heuristic for vehicle routing problems," *Computers & Operations Research*, vol. 39, no. 9, pp. 2033–2050, Sep. 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305054811002826

[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

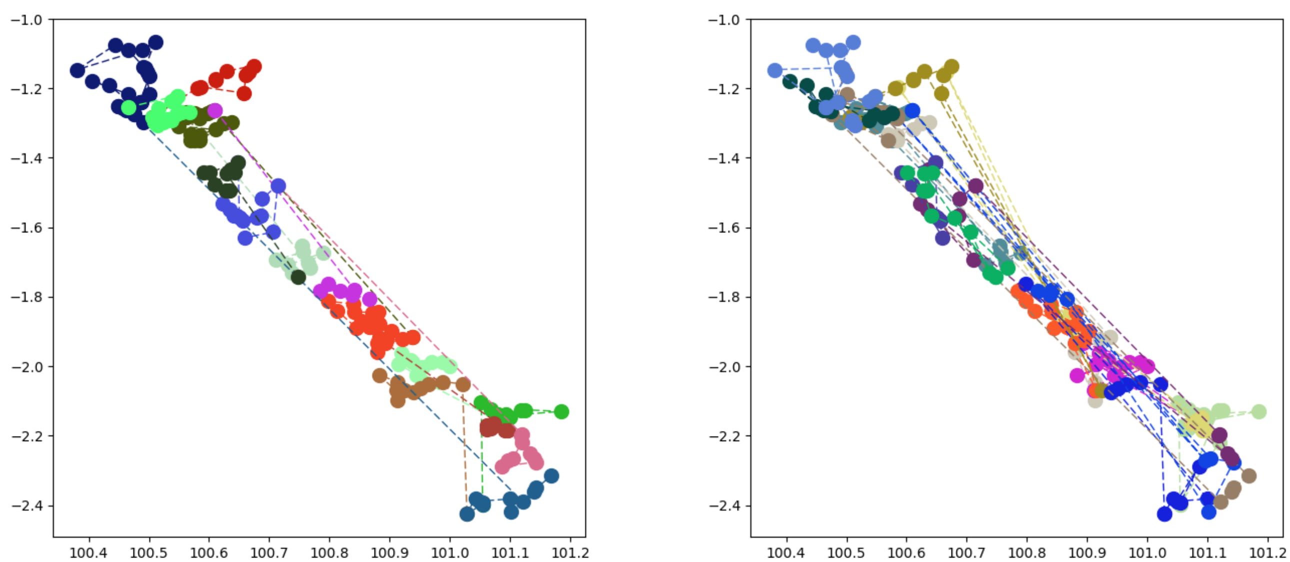
[13] F. B. de Oliveira, R. Enayatifar, H. J. Sadaei, F. G. Guimares, and J.-Y. Potvin, "A cooperative coevolutionary algorithm for the Multi-Depot Vehicle Routing Problem," *Expert Systems with*



(a) MDVRP based on CoEAS

(b) Publish/Subscribe + MDVRP based on CoEAS

Fig. 7: The comparison of routes between MDVRP based on CoEAS and publish/subscribe + MDVRP based on CoEAS for Cordeau data



(a) MDVRP based on CoEAS

(b) Publish/Subscribe + MDVRP based on CoEAS

Fig. 8: The comparison of routes between MDVRP based on CoEAS and publish/subscribe + MDVRP based on CoEAS for real administrative data

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] 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

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