

A Decentralized Coordination Algorithm for a Highly Dynamic Vehicle Routing Problem

Evans Sowah Okpoti · In-Jae Jeong[†]

Department of Industrial Engineering, Hanyang University

동적 차량경로 문제에 대한 분산 알고리즘

이반스 소와 옥포티 · 정인재[†]

한양대학교 산업공학과

The Dynamic Vehicle Routing Problem (DVRP) involves a combinatorial optimization problem where new customer demands become known over time, and old routes must be reconfigured to generate new routes while executing the current solution. We consider the high level of dynamism problem. An application of highly dynamic DVRP is the ambulance service where a patient contacts the service center, followed by an evaluation of case severity, and a visit by a practitioner/ ambulance is scheduled accordingly. This paper considers a variant of the DVRP and proposes a decentralized algorithm in which collaborators (Depot and Vehicle), both have only partial information about the entire system. The DVRP is modeled as a periodic re optimization of VRP using the proposed decentralized algorithm where collaborators exchange local information to achieve the best global objective for the current state of the system. We assume the existence of a dispatcher e.g., headquarter of the company who can communicate to vehicles in order to gather information and assigns the new visits to them. The effectiveness of the proposed decentralized coordination algorithm is further evaluated using benchmark data given in literature. The results show that the proposed method performed better than the compared algorithms which utilize the centralized coordination in 12 out of 21 benchmark problems.

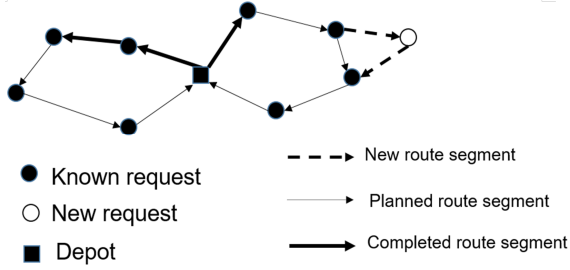
Keywords : Decentralized Coordination, Collaborative Decision-Making, Dynamic Vehicle Routing

1. Introduction

The vehicle routing problem (VRP) involves the assignment a fleet of capacitated vehicles to serve a set of customers in order to minimize the service cost by minimizing the number of vehicles and the total travel distance. The vast majority of VRP research addresses a static problem in which all of the relevant data is known in advance. Typically, ser-

vice vehicles commence at some initial location and all customer demands and constraints are known such as the distance and travel times between each pair of customers and between each customer and the depot. The dynamic VRP (DVRP) can be described as follows : A daily service schedule is predetermined before the start of the working day and drivers start visiting customer with planned routes. During the service, if new customers call for service, routes must be revised. The revised planning must be done such that appropriate vehicles can visit the new customers in efficient ways. <Figure 1> shows an example of a DVRP with 7 known orders and 1 new order. As shown, some customer orders

are already known and an initial route is generated to service these customers. On receipt of new customer order, the existing routes are reconfigured to accommodate the new order. DVRP was first introduced by Kilby et al. [21] and Montemanni et al. [23] introduced some benchmark instances for DVRP.



<Figure 1> DVRP Example

Practical applications of DVRP can be classified depending on the degree of dynamism proposed by [22]. Let T be the length of the planning horizon; the total number of dynamic orders n_{tot} ; \mathbb{R} the set of dynamic orders, and t_i the disclosure time of request $i \in \mathbb{R}$. Time window is the interval $[start(t_i), end(l_i)]$ within which a customer must be serviced. The effective degree of dynamism, δ_{TW}^e is defined as follows :

$$\delta_{TW}^e = \frac{1}{n_{tot}} \sum_{i \in \mathbb{R}} \left(1 - \frac{l_i - t_i}{T} \right)$$

If all the orders are extremely urgent such that $l_i = t_i$, then the effective degree of dynamism becomes 1. On the other hand, if all the orders are not urgent such that $t_i = 0$ and $l_i = T$, δ_{TW}^e becomes 0. Larsen et al. [22] developed three-category framework for DVRP classification namely weakly (low level), moderately (medium level), and strongly dynamic (high level) based on the value of the effective degree of dynamism. The values of δ_{TW}^e according to the levels above are as follows : < 0.3 , between 0.3 and 0.8, and > 0.8 respectively. The cable/ telephone repair service or dial-a-ride/ taxi-cab service that includes door-to-door services with a great number of dynamic customer requests are some-typical examples of low level dynamism problems. A second application is the appliance repair and courier services which involve single/multiple insertions into a planned day's tour with time windows are examples of medium level dynamism problems. A third application is the ambulance service where a patient contacts the service center, followed by an evalua-

tion of case severity, and a visit by a practitioner/ ambulance is scheduled accordingly. This we consider as the high level of dynamism problem.

A typical centralized algorithm for DVRP assumes the existence of a dispatcher e.g., headquarter of the company where the tours are recalculated. The dispatcher can communicate to vehicles in order to gather information and assigns the new visits to them. The decision problem faced by a dispatcher is more complicated than the static VRP which is NP-hard. In addition, due to the uncertain and dynamically changing information at routes and vehicles, such a centralized problem-solving approach is not appropriate to quickly find a high-quality solution for strongly dynamic DVRP. Another barrier for the application of the centralized approach for strongly dynamic DVRP is the applicability of information sharing among the dispatcher and vehicles. In the US, the ambulance service providers are private companies and they are reluctant to reveal the private information of their companies while generating a visiting tour plan with dispatcher (i.e., 911 call center in this case) for urgent patients.

Each entity in the DVRP instance is treated as a collaborator. Each vehicle collaborator knows the information on the customers in its route and exchange information with the depot. A depot collaborator (i.e., dispatcher) is in charge of all the vehicles and knows all the information of old and new customers it is going to serve. Each of these entities has its own objective. The objective of each vehicle collaborator is the traditional minimization of travel time. The objectives of vehicle collaborators are often in competition with each other since the operations of a vehicle collaborator tend to be local based on its local view of the current state. Therefore, there is a need for the depot collaborator who has the global picture of the current state to be able to coordinate the operations of the vehicles. In addition, the objective of the depot collaborator is to minimize the total distance traveled in all the routes.

This paper focuses on developing decentralized cooperation mechanism among vehicle collaborators and the depot collaborator such that the total traveling distance is minimized while the information sharing among collaborators are minimal in a strongly dynamic DVRP environment. The work is organized as follows : in Section 2, we survey previous studies on DVRP. The problem considered in this paper is described in Section 3. The proposed decentralized architecture is described in Section 4. We validate the pro-

posed model via comparison with existing centralized algorithms in Section 5. Finally, some concluding remarks are given in Section 6.

2. Literature Review

In this section, we survey the traditional centralized approaches for DVRP. The model for the dynamic and deterministic VRP can be categorized into two : periodic re-optimization and continuous reoptimization.

The periodic reoptimization models consider initial routes at start of a working day with subsequent reoptimization, either on update of the available data, or at fixed time intervals referred to as time slices [21] or decision epochs [15]. Kilby et al. [21] introduced the specific DVRP model investigated in this research discussing some of the issues created by dynamic problems. They derived benchmark DVRP problems from well-known classic VRP benchmarks. Montemanni et al. [23] refined the benchmarks introduced by [21] and solved the DVRP using a centralized Ant Colony System (ACS). They considered a DVRP as a periodic standard VRP by decomposing a DVRP into a sequence of static VRPs over time slices. Chen and Xu [5] developed a dynamic column generation (DYCOL) approach for the DVRP with time window (DVRPTW) and introduced the concept of decision epochs over a planning period, i.e. dates to run the optimization process. Hanshar and Ombuki-Berman [15] addressed the DVRP using genetic algorithm (GA) and further compared it to a Tabu Search (TS) algorithm. Like [21, 23], they adopted the concept of time slice and evaluated their algorithm based on the benchmark data suggested by [21, 23]. The results showed that the GA outperforms the ACS and TS.

Most recent studies based on the periodic optimization (time slices) scheme that also considered the benchmark data provided by [21, 23] include : Khoudjia et al. [20] applied a dynamic adaptive Particle Swarm Optimization (DAPSO) to the DVRP and further compared the performance of DAPSO to a Variable Neighborhood Search (VNS) algorithm they implemented. Elhassania et al. [10] developed a hybrid algorithm by combining an Ant Colony Optimization (ACO) with a Large Neighborhood Search (LNS) algorithm to solve the DVRP. In their study, a pure ACO was initially adopted as a “control experiment” to the hybrid algorithm (ACOLNS). Demirtas et al. [9] proposed a Particle Swarm Optimization (PSO) which also involved the utilization of

local search procedures and a new path-insertion based heuristic to repair infeasible solutions. Euchi et al. [11] proposed an Artificial Ant Colony based on the 2_Opt local search (AAC_2_Opt) for the DVRP.

Continuous reoptimization models consider optimization throughout the working day while keeping information on good solutions in memory [24]. For any change in available data, a decision procedure collects the information from the memory to update the existing scheduled routes. Note that because the current routing is tentative; vehicles are unaware of their next destination until completion of service of a request [13]. Gendreau et al. [13] applied the parallel Tabu Search framework proposed by [16] to a DVRPTW associated with long distance express courier services. The approach maintains a pool of good routes (in adaptive memory) which are used to generate initial solutions for parallel TS. This framework was also implemented for the DVRP [18, 31]. GAs have also been used for DVRP recently [17]. GAs in dynamic contexts only differ from those designed for static problems in the constant adaptation of solutions to the changes made to the input.

Different variants of DVRP studies exist in the literature. Schilde et al. [25] studied the integration of stochastic time-dependent travel speed in solution methods for the dynamic dial-a-ride problem (DARP). A dynamic nearest neighbor policy for the pick-up and delivery problem with multiple vehicles was introduced by [26]. The policy is for the operation of a fleet of vehicles used to serve customers within a Euclidean service area according to a Poisson process. Yang et al. [32] also studied a multi-objective distribution model that considers the customers’ satisfaction degree as well as cost for online shopping express logistics and proposed a modified PSO which enhances the particle evolution quality. Dan et al. [8] considered the Emergency Materials Dispatch (EMD) as a typical DVRP and developed a model as well as an ACO algorithm to solve the problem. Barkaoui et al. [2] introduced an adaptive evolutionary approach that uses genetic algorithm for real-time vehicle routing and dispatching. Azi et al. [1] proposed a method based on an adaptive large neighborhood search heuristic to solve a DVRP with multiple delivery routes. Kergosien et al. [19] proposed a tabu search heuristic for the transportation of patients between care units dynamically and Beaudry et al. [3] introduced a two-phase heuristic for the dynamic transportation of patients in hospitals. Bent et al. [4] developed a LNS strategy for the DVRP and also considered a multiple

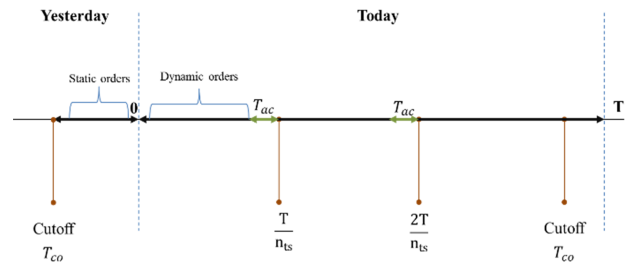
scenario approach, which uses past decisions to generate plans. Ichoua et al. [17] studied a similar problem to that in [4] by exploiting information on future events to improve decision making. Zhu et al. [33] considered a real-time DVRP and applied open constraint programming. Coslovich et al. [7] considered an urban dial-a-ride problem and proposed a two-phase insertion algorithm. See Gendreau et al. [13] for other neighborhood search heuristics for dial-a-ride problem.

In this research, we study the periodic reoptimization model of DVRP with a decentralized coordination using decentralized system hierarchy.

3. Problem Description

As shown in <Figure 2>, the periodic reoptimization model of DVRP discussed in this research, also studied by [21, 23], can be described as follows : T_{co} is a cut-off time like real world scenarios after which requests arriving are postponed to the next working day. Thus the dynamic orders that arrived in the time window $[-T_{co,0}]$ is called **pending orders**. At time $t = 0$, a static VRP is solved. The static VRP can be solved with an assumption that all the relevant information of orders and vehicles are gathered and centered to the depot. This assumption can be applied to both the centralized and decentralized coordination environments. It is reasonable to assume that the relevant information on vehicles are available to the depot since all the vehicles are located at the depot at the beginning of the day. Then the starting time of the service of the **pending orders** are determined and the orders are transformed to **planned order**. Among the planned orders that can be serviced within $\left[0, \frac{T}{n_{ts}} + T_{ac}\right]$ is called **committed order**. n_{ts} is the number of time slices. Thus $\frac{T}{n_{ts}}$ represents the length of a time slice. T_{ac} is an advanced commitment time. In practice, an order must be committed to a vehicle at least T_{ac} seconds before departure from the last visited location prior to start of service of the committed order. This advanced commitment time gives an appropriate time to react to committed new orders. The vehicle to which a committed order is assigned must leave from its previous customer (last service location). At $t = \frac{T}{n_{ts}}$, there is a planned order, that is **pending orders** \ **committed orders** (i.e., pending orders except committed orders)

and in addition we have newly arrived **dynamic orders** during the period $\left[0, \frac{T}{n_{ts}}\right]$. Therefore, at the beginning of this new time slice, we have a new static VRP again. In a centralized coordination environment, we assume that vehicles serve customers along with the planned route, send all the relevant information of its own (i.e., current location of vehicle, current available capacity and so on). However, in this research, we assume only a decentralized coordination is applicable (See section 4 for more details) to solve the static VRP where minimal information is shared among collaborators and decision authority is dispersed among collaborators. That is, there is no decision maker like a dispatcher who has a global view of the entire static VRP. At this point, the decentralized algorithm is applied again for orders, **pending order** \ **committed order** + **dynamic order**. Among the newly planned orders, serviceable orders during $\left[\frac{T}{n_{ts}} + T_{ac}, \frac{2T}{n_{ts}} + T_{ac}\right]$ can be committed and this procedure repeats until the current time reaches T_{co} of today.



<Figure 2> Periodic Reoptimization DVRP Model

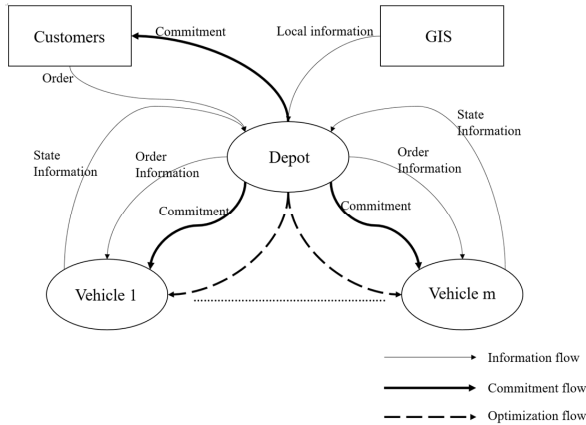
4. Decentralized Architecture

<Figure 3> shows the proposed decentralized architecture among depot collaborator and vehicle collaborators. Collaborators are independent and intelligent entities that are loaded with some programs e.g., heuristic algorithm or optimization software such as CPLEX, and can therefore react to events and perform some jobs.

The decentralized interaction is characterized by the optimization stage and the commitment stage. In the optimization stage, the depot collaborator and vehicle collaborators cooperatively solve the reoptimization problem of DVRP. Throughout the time horizon, relevant information regarding received customer orders in a time slice such as customer's geographic location and traveling distances are verified with GIS by the depot collaborator. The depot collaborator sends

information on customers to each vehicle collaborator. The vehicle collaborator on the other hand sends its state information such as current location of vehicle and planned route to depot collaborator. With this information, depot collaborator re-optimizes the vehicle routes by solving its own optimization problem considering customers and vehicle collaborators' information. Next, customer-vehicle assignments are announced to vehicle collaborators. Once a vehicle collaborator has received information on new customers, it determines a routing plan by solving its optimization problem and sends the information back to the depot collaborator. This iterative process continues while there is ample time until the best objective value is reached.

Once the optimization terminates, orders are committed to specific vehicles at commitment stage by the depot collaborator. Note that orders must be committed to a driver at least an advanced commitment time, T_{ac} prior to departure from the last location visited. The advanced commitment time gives the drivers appropriate reaction time to new orders before commencement of processing the order. The proposed architecture is purely decentralized since neither the depot or vehicle collaborator has a global view of the entire state of the system. However, this can be compensated by the information exchange among the collaborators.



<Figure 3> Decentralized Architecture

The proposed decentralized procedure can be formalized as follows :

Phase I

Step 0 : Set the current time $t = 0$

Step 1 (Optimization stage) : Let P be the set of pending orders at $t = 0$. The depot collaborator solves the static VRP with nodes, P . Let P_k be the subset of pending orders as-

signed to vehicle k and R_k be the set of pair of orders which represent the corresponding visiting sequence of orders. Let V be the set of vehicles that are assigned orders.

Step 2 (Commitment stage) : Among P_k , commit the set of orders, C_k that can be served within the time window, $\left[0, \frac{T}{n_{ts}} + T_{ac}\right]$ along with the traveling sequence R_k .

Step 3 : Let the orders that have not been served within the time window $\left[0, \frac{T}{n_{ts}} + T_{ac}\right]$ be unplanned orders H_k , which are the set of pending orders\committed orders assigned to vehicle collaborator k , that is $P_k \setminus C_k$. Also let D be the set of dynamic orders received during the time slice $\left[0, \frac{T}{n_{ts}}\right]$. Set the current time $t = \frac{T}{n_{ts}}$.

Phase II

Step 4 (Optimization stage)

Step 4.1 (Initial Solution) : The depot collaborator solves the Linear Assignment Problem (LAP) for $|D|$ objects (i.e., dynamic orders) and $|V|$ persons (i.e., vehicles) with the assignment cost $c_i^k = \min_{j \in H_k} [d_{ij}]$ for all $i \in D$ when a dynamic order i is assigned to a vehicle k .

Step 4.2 (Vehicle collaborator) : The vehicle collaborator k solves the Vehicle collaborator Problem VAP (C_k, R_k, v_i^k, H_k) . As a result, the vehicle collaborator k updates R_k such that the order v_i^k is newly inserted. In addition, the existing R_k for H_k is newly optimized given that orders in C_k are already served. The vehicle collaborator k sends the local objective function value $f(v_i^k)$ and unplanned orders, H_k to the depot collaborator.

Step 4.3 (Improvement stage) : The depot collaborator gathers local objective function values from vehicle collaborators and updates the upper bound, $\sum_{k=1}^m f(v_i^k)$. Set $\alpha = 1$

Step 4.3.1 (Depot collaborator) : Randomly select orders from $D \cup H$ and generate a sequence of orders. Let $i_{[j]}$ be the j^{th} element in the sequence. Set $j = 1$. For $i_{[j]}$, let $v_{i_{[j]}}^k = 1$. The depot collaborator sends the assignment to the vehicle collaborator k .

Step 4.3.2 (Vehicle collaborator) : The vehicle collaborator k solves VAP $(C_k, R_k, v_{i_{[j]}}^k, H_k)$ and send the corresponding objective function value, $f(v_{i_{[j]}}^k)$ to the depot collaborator.

Step 4.3.3(Depot collaborator) : Select the vehicle collaborator whose $f(v_{i[j]}^k)$ is minimal among all vehicle collaborators and assign the order $i[j]$ to the corresponding vehicle collaborator.

Step 4.3.4 : If $i[j] \neq \alpha|D \cup H|$ increase $j = j + 1$ and go to Step 4.3.1.

Step 4.3.5 : If the new objective function value $\sum_{k=1}^m f(v_{i[j]}^k)$ has been improved from the current upper bound, update the current upper bound as $\sum_{k=1}^m f(v_{i[j]}^k)$, set $\alpha = 0.3$ and go to Step 4.3.1. Otherwise terminate the algorithm.

Define LAP as follows :

$$v_i^k = \begin{cases} 1 & \text{vehicle } k \text{ serves customer } i \\ 0 & \text{otherwise} \end{cases}$$

LAP

$$\text{Minimize} \quad \sum_{k=1}^m \sum_{i=1}^{|D \cup H|} c_i^k \quad (1)$$

$$\text{s.t} \quad \sum_{k=1}^m v_i^k = 1, \quad i \in \{D \cup H\} \quad (2)$$

$$\sum_{i=1}^{|D \cup H|} v_i^k = 1 \quad k = 1, \dots, m \quad (3)$$

Step 5 (Commitment stage)

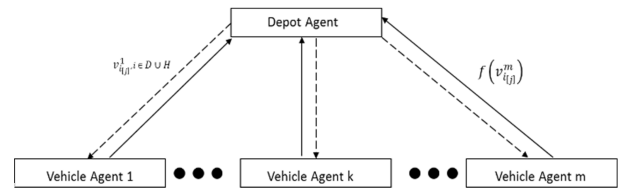
Step 5.1 : For each vehicle collaborator k , commit orders in D_k and H_k along with the route in R_k where the service can be completed within the next time slice plus T_{co} .

Step 5.2 : Increase the current time by $\frac{T}{n_{ts}}$. If the time slice reaches to the cutoff time T_{co} , terminate the algorithm otherwise, go to Step 1.

Phase I deals with the optimization and commitment stage at $t = 0$ and Phase II applies at the beginning of each time slices such as $\frac{T}{n_{ts}}, \frac{2T}{n_{ts}}, \frac{3T}{n_{ts}}$ and so on. The static VRP in Step 1 is heuristically solved using the Ruin and Recreate (R&R) algorithm proposed by Schrimpf et al. [27]. The R&R algorithm has been successfully applied to VRPTW and outperformed existing meta heuristics (see [27] for more details). Once the algorithm at $t = 0$ terminates, the committed orders are served based on the route determined from the static VRP at Step 1.

In order to solve LAP at step 4.2, $|D|$ must be the same as $|V|$. When $|D| > |V|$, we add $|D| - |V|$ new vehicles to the solution in order to handle new orders. In this case, c_i^k are the same for all new vehicles since all the vehicles are located at the depot. If $|D| < |V|$, we add $|V| - |D|$ artificial orders with $c_i^k = 0$ for all vehicles so that we have the equal number of orders and vehicles. The LAP is solved optimally using the well-known Hungarian algorithm. The vehicle collaborator problem, $VAP(C_k, R_k, v_k^k, H_k)$ is formalized in the following section.

Step 4.3 deals with the improvement of the initial solution by the decentralized cooperation among the depot collaborator and vehicle collaborators. Upon receipt of an upper bound of solution, the depot collaborator resets the assignment of the unplanned orders and dynamic orders to vehicle collaborators which was determined at Step 3 and Step 4.1 respectively. The depot collaborator tries to improve the initial solution by solving the General Assignment Problem (GAP) of $D \cup H$ objects to K persons where K is the set of uniform fleet of vehicles with same capacity at Step 4.3. Different to LAP, we allow the assignment of multiple orders to one vehicle (i.e. relaxing constraint 2) in GAP which is NP-hard problem. Actually, Step 4.3 shows the heuristic and decentralized algorithm to solve the GAP of the depot collaborator. The information flows among the depot collaborator and the vehicle collaborators for Step 4.3 are shown in <Figure 4> At Step 4.3.1, the depot collaborator asks the vehicle collaborator k to send the local objective function value when the vehicle covers the order, $v_{i[j]}^k$ in its current route of visiting customers.



<Figure 4> Information Flows between the Depot Collaborator and the Vehicle Collaborator

Upon receipt of all the local objective function value, the depot collaborator select the most efficient vehicle (i.e., the lowest objective function value) and assign $v_{i[j]}^k$ to the corresponding vehicle. The procedure continues until all the orders in $D \cup H$ are assigned to vehicles. Step 4.3 is repeated as long as the computation time of algorithm does not exceed

$\frac{T}{n_{ts}}$ and the best solution is implemented for the next time slice.

The optimization problem of vehicle collaborators VAP ($C_k, R_k, v_{i[j]}^k, H_k$) can be formalized as follows :

Parameters

q_i : weight (or volume) of customer i

Q : capacity of vehicle

Decision variables

$x_{ij}^k = \begin{cases} 1: & \text{vehicle } k \text{ departs from customer } i \text{ to } j, \forall k \\ 0: & \text{otherwise} \end{cases}$

$v_i^k = \begin{cases} 1: & \text{vehicle } k \text{ serves customer } i \text{ to } j, \forall k \\ 0: & \text{otherwise} \end{cases}$

VAP($C_k, R_k, v_{i[j]}^k, H_k$)

$$f(R_k) = \text{Minimize} \sum_{i \in v_{i[j]}^k \cup H_k} \sum_{j \in v_{i[j]}^k \cup H_k} d_{ij} x_{ij}^k \quad (4)$$

Subject to

$$\sum_{j \in v_{i[j]}^k \cup H_k, j \neq i} x_{ij}^k = 1, \quad j \in v_{i[j]}^k \cup H_k \quad (5)$$

$$\sum_{j \in v_{i[j]}^k \cup H_k, i \neq j} x_{ij}^k = 1, \quad i \in v_{i[j]}^k \cup H_k \quad (6)$$

$$\sum_{i \in A} \sum_{i \in A, i \neq j} x_{ij}^k \leq |A| - 1, \quad A \subset C_k \cup v_{i[j]}^k \cup H_k, \quad (7)$$

$$2 \leq |A| \leq \frac{|C_k \cup v_{i[j]}^k \cup H_k|}{2} \quad (7)$$

$$\sum_{i, j \in C_k \cup v_{i[j]}^k \cup H_k} q_i x_{ij}^k \leq Q, \quad (8)$$

$$x_{ij}^k = 1, \quad (i, j) \in R_k, \quad i, j \in C_k \quad (9)$$

$$x_{ij}^k \in \{0, 1\} \quad i, j \in v_{i[j]}^k \cup H_k \quad (10)$$

The objective function of VAP is the minimization of total traveling distance in order to serve orders that are assigned by the depot collaborator and unplanned order that was pending order from the previous day. The constraint (5) and (6) imply that the order must be served once and must be connected as a route. The subtours must be eliminated by the constraint (7). The constraint (8) represents the vehicle capacity. The constraint (9) implies that the visiting sequence of already

committed orders must be maintained. Note that the existing route of the vehicle R_k is updated only for $v_{i[j]}^k \cup H_k$.

The proposed VAP is a constrained VRP where the visiting sequence of some customers is predetermined. We heuristically solve the VAP using the R&R algorithm.

5. Experimental Results

This section presents the experiment conducted and the corresponding results. The decentralized DVRP system was coded in Eclipse 4.4 (Luna) and ran on Intel(R) Core(TM) i5 CPU (760 @ 2.80GHz (4 CPUs), ~2.8GHz) with 4096MB memory. A comparison of the decentralized system's performance with other algorithms is also presented.

5.1 Benchmark Dataset

As mentioned earlier, the experimental results are based on the instances proposed by [21, 23]. They were derived from three separate already known VRP benchmark data by Taillard [29] (12 instances), Christophides and Beasley [6] (7 instances) and Fisher [12] (2 instances). These instances were extended by [21] and organized into two group of instances, pickup and delivery. This research considered only pickup problems (total of 21 instances) which include the following information :

- length of the working day. Referred to as T.
- appearance time of each order. This denotes the time when the order becomes known to the dispatcher.
- duration of each order. It represents the service time for each order.
- number of vehicles. It represents the number of vehicles available for serving the customers (set to 50).

Hanshar and Ombuki-Berman [15] further analyzed the topology of the service areas for the benchmark datasets shown in <Table 1>. Generally, the service areas are either uniformly distributed, clustered or a mixture of the two.

<Table 1> Topology Details of the Benchmark Dataset

Type	Customers	Distribution
<i>f</i> -series	71~134	Center-clustered
<i>Tai</i> -series	75~150	Clustered, Uniform and Mixed
<i>c</i> -series	50~199	Clustered and Uniform

In order to compare our results with the other mentioned algorithms, the same experimental settings of [15] has been adopted. The number of time slices in the optimization, $n_{ts} = 25$. Also, the advanced commitment time T_{ac} was set at $0.01T$, where T is the entire length of the working day. Also, the cutoff time T_{co} was set at $0.5T$. 30sec is allowed for the running of the algorithm in each time slice. If the algorithm did not complete within 30sec, the best solution found is saved and the subsequent time step begins. Thus a running time of the entire algorithm for a single problem instance is $30 \times 25 = 750s$ or 12.5 min for simulating an 8-hour day.

5.2 Comparison With Other Algorithms

<Table 2> presents the numerical results for the decentralized compared to other centralized DVRP algorithms such as dynamic adaptive Particle Swarm Optimization (DAPSO) and Variable Neighborhood Search (VNS) by [7], a hybrid Ant Colony Optimization-Large Neighborhood Search (ACOLNS) and ACO by [8] and Artificial Ant Colony with 2_Opt (AAC_2_Opt) by [10].

For each of the centralized algorithms, the objective function values reported is the average value reported by the cor-

responding author. The results show that the decentralized system comparatively achieved the best solution for 12 out of 21 instances. The AAC_2_Opt recorded 9 best solutions out of the 21 instances.

It seems that the proposed decentralized approach performs especially well when the number of customers is large (i.e., more than 100 customers).

6. Conclusion

In this research, DVRP has been studied. Each entity in the DVRP instance is treated as a collaborator having its own local objective that partially represents the global system objective. The collaborators can execute a set of operations to change their status to align with the overall system objective.

We proposed a decentralized architecture and coordination algorithm for DVRP. Also the proposed method has been tested using problem instances derived from publicly available benchmark data. The results show that the proposed method performed better than AAC_2_Opt based method which utilizes the centralized coordination in 12 out of 21 benchmark problems respectively.

<Table 2> Algorithm Comparison

Instance	MA	DAPSO (Average)	VNS (Average)	ACOLNS (Average)	ACO (Average)	AAC_2_Opt (Average)
c50	691.87	632.38	653.84	623.09	659.60	570.89
c75	1007.49	1031.76	1040.00	1013.47	1101.03	1213.45
c100	996.72	1051.50	1087.18	1012.30	1120.31	1380.25
c100b	898.36	964.47	942.81	943.05	972.30	841.44
c120	1289.64	1457.22	1469.24	1451.60	1481.27	1153.29
c150	1347.52	1470.95	1441.37	1394.77	1559.00	2386.93
c199	1640.41	*	1769.95	1757.02	1892.48	1758.51
f71	323.46	312.15	325.18	320.00	358.45	309.94
fl34	14280.54	*	16522.18	16030.53	17983.84	15528.81
tai75a	1768.80	1935.28	1954.25	1880.87	2215.75	1782.91
tai75b	1455.72	1484.73	1560.71	1477.15	1821.00	1452.26
tai75c	1511.29	1664.40	1746.07	1692.00	2076.19	1441.91
tai75d	1496.67	1664.40	1541.98	1491.84	1748.09	1422.27
tai100a	2191.72	2370.58	2462.50	2331.28	2863.71	2232.71
tai100b	2165.28	2385.54	2319.72	2317.30	2750.01	2182.61
tai100c	1510.85	1627.32	1557.81	1717.61	2051.87	1562.66
tai100d	1894.29	2123.90	2100.38	2087.96	2410.37	1912.43
tai150a	3473.61	*	3680.35	3595.40	4191.37	3185.73
tai150b	2944.52	*	3089.57	3095.61	3775.38	2880.57
tai150c	2618.08	*	2928.77	2840.69	3733.77	2743.55
tai150d	2940.76	*	3147.38	3233.39	3745.31	3345.16

Bold values indicate the best among all algorithms
 * : means value is not available

In near future, we will consider the stability of VRP plan as the local objective of a vehicle collaborator. With the revised route, we may ignore the current VRP plan and accept a new plan which is completely different route from the old one. However, driver is usually nervous about the change in the route plan. A driver may be reluctant to have changes in lunch break due to an inserted customer in his/her route. Therefore, from the optimization point of view, it is reasonable to accept significant changes in the route. However, from the execution point of view, the stability of route may be more important than the optimization. This way, one can observe some learning capabilities of collaborators and further design artificial intelligence approaches.

Acknowledgement

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea, funded by the Ministry of Education (grant number 2014 R1A1A2058147) and the Ministry of Science, ICT and Future Planning (grant number 2017R1E1A1A03070435).

References

- [1] Azi, N., Gendreau, M., and Potvin, J.Y., A dynamic vehicle routing problem with multiple delivery routes, *Annals of Operations Research*, 2012, Vol. 199, No. 1, pp. 103-112.
- [2] Barkaoui, M. and Gendreau, M., An adaptive evolutionary approach for real-time vehicle routing and dispatching, *Computers and Operations Research*, 2013, Vol. 40, No. 7, pp. 1766-1776.
- [3] Beaudry, A., Laporte, G., Melo, T., and Nickel, S., Dynamic transportation of patients in hospitals, *OR Spectrum*, 2010, Vol. 32, No. 1, pp. 77-107.
- [4] Bent, R.W. and Van Hentenryck, P., Scenario-based planning for partially dynamic vehicle routing with stochastic customers, *Operations Research*, 2004, Vol. 52, No. 6, pp. 977-987.
- [5] Chen, Z.-L. and Xu, H., Dynamic column generation for dynamic vehicle routing with time windows, *Transportation Science*, 2006, Vol. 40, No. 1, pp. 74-88.
- [6] Christofides, N. and Beasley, J.E., The period routing problem, *Networks*, 1984, Vol. 14, No. 2, pp. 237-256.
- [7] Coslovich, L., Pesenti, R., and Ukovich, W., A two-phase insertion technique of unexpected customers for a dynamic dial-a-ride problem, *European Journal of Operational Research*, 2006, Vol. 175, No. 3, pp. 1605-1615.
- [8] Dan, B., Zhu, W., Li, H., Sang, Y., and Liu, Y., Dynamic optimization model and algorithm design for emergency materials dispatch, *Mathematical Problems in Engineering*, 2013.
- [9] Demirtas, Y.E., Ozdemir, E., and Demirtas, U., A particle swarm optimization for the dynamic vehicle routing problem, In Modeling, Simulation, and Applied Optimization (ICMSAO), *2015 6th International Conference on IEEE*, 2015, pp. 1-5.
- [10] Elhassania, M.J., Jaouad, B., and Ahmed, E.A., A New Hybrid Algorithm to Solve the Vehicle Routing Problem in the Dynamic Environment, *International Journal of Soft Computing*, 2013, Vol. 8, pp. 327-334.
- [11] Euch, J., Yassine, A., and Chabchoub, H., The dynamic vehicle routing problem : Solution with hybrid meta-heuristic approach, *Swarm and Evolutionary Computation*, 2015, Vol. 21, pp. 41-53.
- [12] Fisher, M.L. and Jaikumar, R., A generalized assignment heuristic for vehicle routing, *Networks*, 1981, Vol. 11, No. 2, pp. 109-124.
- [13] Gendreau, M., Guertin, F., Potvin, J.-Y., and Seguin, R., Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries, *Transportation Research Part C : Emerging Technologies*, 2006, Vol. 14, No. 3, pp. 157-174.
- [14] Gendreau, M., Guertin, F., Potvin, J.-Y., and Taillard, E., Parallel tabu search for real-time vehicle routing and dispatching, *Transportation Science*, 1999, Vol. 33, No. 4, pp. 381-390.
- [15] Hanshar, F.T. and Ombuki-Berman, B.M., Dynamic vehicle routing using genetic algorithms, *Applied Intelligence*, 2007, Vol. 27, pp. 89-99.
- [16] Ichoua, S., Gendreau, M., and Potvin, J.-Y., Diversion issues in real-time vehicle dispatching, *Transportation Science*, 2000, Vol. 34, No. 4, pp. 426-438.
- [17] Ichoua, S., Gendreau, M., and Potvin, J.Y., Exploiting knowledge about future demands for real-time vehicle dispatching, *Transportation Science*, 2006, Vol. 40, No. 2, pp. 211-225.
- [18] Ichoua, S., Gendreau, M., and Potvin, J.-Y., Vehicle dispatching with time-dependent travel times, *European Journal of Operational Research*, 2003, Vol. 144, No. 2, pp. 379-396.

- [19] Kergosien, Y., Lente, C., Piton, D., and Billaut, J.C., A tabu search heuristic for the dynamic transportation of patients between care units, *European Journal of Operational Research*, 2011, Vol. 214, No. 2, pp. 442-452.
- [20] Khouadjia, M.R., Sarasola, B., Alba, E., Jourdan, L., and Talbi, E.G., A comparative study between dynamic adapted PSO and VNS for the vehicle routing problem with dynamic requests, *Applied Soft Computing*, 2012, Vol. 12, No. 4, pp. 1426-1439.
- [21] Kilby, P., Professor, P., and Shaw, P., Dynamic VRPs : a study of scenarios, Technical report APES-0-1998, University of Strathclyde, 1998.
- [22] Larsen, A., Madsen, O., and Solomon, M., Partially Dynamic Vehicle Routing-Models and Algorithms, *The Journal of the Operational Research Society*, 2002, Vol. 53, No. 6. pp. 637-646.
- [23] Montemanni, R., Gambardella, L.M., Rizzoli, A.E., and Donati, A.V., Ant colony system for dynamic vehicle routing problem, *Journal of Combinatorial Optimization*, 2005, Vol. 10, pp. 327-343.
- [24] Pillac, V., Gendreau, M., Gueret, C., and Medaglia, A.L., A review of dynamic vehicle routing problems, *European Journal of Operational Research*, 2013, Vol. 225, No. 1, pp. 1-11.
- [25] Schilde, M., Doerner, K.F., and Hartl, R.F., Integrating stochastic time-dependent travel speed in solution methods for the dynamic dial-a-ride problem, *European Journal of Operational Research*, 2014, Vol. 238, No. 1, pp. 18-30.
- [26] Sheridan, P.K., Gluck, E., Guan, Q., Pickles, T., Balciog, B., and Benhabib, B., The dynamic nearest neighbor policy for the multi-vehicle pick-up and delivery problem, *Transportation Research Part A : Policy and Practice*, 2013, Vol. 49, pp. 178-194.
- [27] Shrimpf, G., Schneider, J., Stamm-Wilbrandt, H., and Dueck, G., Recording breaking optimization results using the ruin and recreate principle, *Journal of Computational Physics*, 2000, Vol. 159, pp. 139-171.
- [28] Taillard, E., Badeau, P., Gendreau, M., Guertin, F., and Potvin, J.-Y., A tabu search heuristic for the vehicle routing problem with soft time windows, *Transportation Science*, 1997, Vol. 31, No. 2, pp. 170-186.
- [29] Taillard, E., Parallel iterative search methods for vehicle routing problems, *Networks*, 1993, Vol. 23, No. 8, pp. 661-673.
- [30] Taillard, E.D., Gambardella, L.M., Gendreau, M., and Potvin, J.-Y., Adaptive memory programming : A unified view of metaheuristics, *European Journal of Operational Research*, 2001, Vol. 135, No. 1, pp. 1-16.
- [31] Van Hemert, J.I. and La Poutre, J.A., Dynamic routing problems with fruitful regions : Models and evolutionary computation, in *Parallel Problem Solving from Nature-PPSN VIII*, 2004.
- [32] Yang, J., Li, J., Chen, Y., and Liu, X., Multi-Objective Distribution Model and Algorithm for Online Shopping Express Logistics, *Journal of Computers*, 2013, Vol. 8, No. 10, pp. 2558-2564.
- [33] Zhu, K.Q. and Ong, K.-L., A reactive method for real time dynamic vehicle routing problem, in *2000 IEEE 24th International Conference on Tools with Artificial Intelligence*, IEEE Computer Society, 2000.

ORCID

Evans Sowah Okpoti | <http://orcid.org/0000-0002-7776-7950>

In-Jae Jeong | <http://orcid.org/0000-0001-7824-6391>