

A-Static-Periodic Solution Strategy for Dynamic Vehicle Routing Problem with Simultaneous Pickup and Delivery

Dinamik Eş Zamanlı Toplamalı ve Dağıtım Araç Rotalama Problemi için Statik Periyodik Çözüm Stratejisi

Mustafa Demirbilek¹ 



ABSTRACT

Nowadays, logistic companies are dealing with the pressure of a highly competitive environment and trying to reduce operational costs. At the same time, customers are requiring not to wait so long to get their delivery and pickup demands satisfied. In this study, these two factors are considered with a-static-periodic solution strategy. The problem is modelled as Dynamic Vehicle Routing Problem with Simultaneous Pickup and Delivery (DVRPSPD) where delivery and pickup demands of customers continuously arrived in a planning horizon are satisfied at the same time and by the same vehicle. Furthermore, we proposed a solution methodology based on solving routing problems repeatedly for each time period by considering new and unvisited previous customers. A well-known heuristic, the Nearest Neighbourhood Algorithm (NNA), is used to solve VRPSPDs in each time period. Important real-life aspects such as demand fluctuations, different number of customers, and routing periods are integrated into the problem. According to results, the average waiting times per customer significantly increase whereas average travel times per visit notably decrease when the length of time period increases. If routes are constructed 8 times in a day, each customer waits an average 5 minutes compared to 66 minutes if it is made 4 times in the day. More routing plans cause more than 7-minute average travel times per visit. Different test settings are analysed and the results are explained to help decision makers find the best solution for both companies and customers.

Keywords: Vehicle routing problem with simultaneous pickup and delivery, The nearest neighbourhood algorithm, Dynamic customer arrivals, Decision making

ÖZ

Günümüzde lojistik firmaları yüksek rekabet ortamının getirdiği baskılarla mücadele etmekte ve operasyonel maliyetleri düşürmeye çalışmaktadırlar. Aynı zamanda müşteriler, dağıtım ve toplama taleplerinin karşılanması için çok beklemek istememektedirler. Bu çalışmada, bahsedilen iki faktör statik periyodik çözüm stratejisiyle beraber ele alınmıştır. Problem, bir planlama dönemi içinde sürekli olarak gelen müşterilerin, toplama ve dağıtım taleplerinin aynı zamanda ve araçla karşılandığı “Dinamik Eş Zamanlı Toplamalı ve Dağıtım Araç Rotalama Problemi (DEZTDARP)” olarak modellenmiştir. Bununla beraber yeni ve ziyaret edilmemiş eski müşterileri göz önüne alarak rotalama problemlerini her zaman periyodu için tekrar tekrar çözen bir çözüm metodu önerilmiştir. Literatürde çok kullanılan bir sezgisel olan En Yakın Komşu Algoritması, her periyotta EZTDARP’lerini çözmek için kullanılmıştır. Talep dalgalanması, farklı sayıda müşteri, farklı rotalama periyotları gibi gerçek hayatta karşılaşılabilecek önemli hususlar probleme entegre edilmiştir. Sonuçlara bakıldığında, rotalama periyotlarının süresi arttığı zaman müşteri başı ortalama bekleme sürelerinin ciddi şekilde arttığı, ziyaret başı ortalama seyahat sürelerinin ise azaldığı görülmüştür. Rotalama planları günde 8 sefer yapıldığında müşteri başı ortalama bekleme süresi 5 dakika iken, plan sayısı 4 olduğunda bekleme süresi 66 dakikaya yükselmektedir. Daha fazla rotalama ise ziyaret başı fazladan 7 dakika ortalama seyahat süresine neden olmaktadır. Karar vericilerin hem firmalar hem de müşteriler için en iyi çözümü bulmalarına yardımcı olmak amacıyla farklı testler analiz edilmiş ve sonuçları açıklanmıştır.

Anahtar kelimeler: Eş zamanlı toplamalı ve dağıtım araç rotalama problemi, En yakın komşu algoritması, Dinamik müşteri gelişleri, Karar verme

¹(Asst. Prof.), Gaziantep Islamic Science and Technology University, Faculty of Engineering And Natural Sciences, Department of Industrial Engineering, Gaziantep, Turkey

ORCID: M.D. 0000-0002-1520-2882

Corresponding author:

Mustafa DEMİRBILEK
Gaziantep Islamic Science and Technology University, Faculty of Engineering And Natural Sciences, Department of Industrial Engineering, Gaziantep, Turkey

E-mail address:
mustafa.demirbilek@gibtu.edu.tr

Submitted: 26.11.2020

Revision Requested: 26.04.2021

Last Revision Received: 02.05.2021

Accepted: 31.05.2021

Published Online: 07.06.2021

Citation: Demirbilek, M. (2021). A-static-periodic solution strategy for dynamic vehicle routing problem with simultaneous pickup and delivery. *Acta Infologica*, 5(1), 1-12. <https://doi.org/10.26650/acin.831973>

1. INTRODUCTION

Vehicle Routing Problem (VRP) can be simply defined as finding the shortest route between a depot and geographically scattered points by satisfying a set of constraints. VRP has attracted many researchers' attention since the study of Dantzig and Ramster (1959), "The Truck Dispatching Problem." They aimed to satisfy demands of fuel stations when minimizing total travel times of a given truck fleet. A linear programming approach was used to solve the problem. Since then, many different types of VRP such as VRP with Time Windows, Stochastic VRP, Multi Depot VRP, Periodic VRP, Dynamic VRP, and different combinations of these have been studied. Not only problem types are evolved, but also a variety of solution methodologies have been developed. Exact solution methods, such as Branch-and-Bound Algorithms (Christofides & Eilon, 2006), Dynamic Programming (Columbus, 1972), Set Partitioning Formulations and Algorithms, and Commodity Flow Formulations and Algorithms, and heuristic/metaheuristic solution methods, such as The Savings Algorithm, Cluster-First-Route-Second Heuristics, Local Search, and Population Search algorithms, have been proposed over 50 years (Laporte, 2009).

VRP with simultaneous pickup and delivery (VRPSPD) is getting attention among other VRPs due to scarce resources, environmental sensitivities, and regulations (Demirbilek, 2020). In VRPSPD, vehicles deliver as well as collect goods at the same time. Each day in the planning horizon, a given number of vehicles start their travels from a depot, deliver and pick up goods to/from customers by visiting them only once, and return to the depot at the end of the day. As classical VRPs, the aim is to minimise total distances as well as to maximise total number of visited customers by considering the capacity of each vehicle (Goksal et al., 2013; Chen & Wu, 2006). The online order system of grocery stores is a good example for VRPSPD. Orders are distributed to customers while the same vehicles pick up undated goods or empty bottles. Because of regulations, some companies are responsible for products during their whole life cycles (as in the disposal of laser printer cartridges) (Montané & Galvão, 2006).

On the other hand, the dynamism issue that considers the arrival of customers over a planning time and assigns them to a schedule as soon as possible is becoming a more and more important issue since the hectic business life and the unwillingness of customers to wait for vehicles to get their packages delivered or picked up. Therefore, the number of studies about Dynamic VRP as well as VRPSPD has significantly increased recently.

In this study, a static-periodic-resolution-based solution methodology for a Dynamic VRPSPD (DVRPSPD) is proposed. The idea behind the solution methodology is to solve VRPSPD for a prespecified time period by covering new customers and previous customers that are not served until that time period. Therefore, customer arrivals during a planning horizon are taken into consideration in a dynamic fashion instead of optimizing routes with known customers before the planning horizon starts as in static solution approaches in the literature. A well-known heuristic algorithm, the Nearest Neighbourhood Algorithm (NNA), is used to solve VRPSPD in each period. The main contribution of the study is to examine relationships between average travel times for vehicles and the waiting times for customers under consideration of different routing period lengths and demand fluctuations.

In the next section, recent studies in the literature are reviewed. In Section 3, the mathematical formulation and NNA are explained. In Section 4, experimental setting, necessary data, and results are described. In the last section, we conclude our study and emphasize some assumptions and limitations.

2. LITERATURE REVIEW

Although many studies about DVRP and VRPSPD exist in the literature, it was observed that only the study of Aydogdu and Ozyoruk (2020) considered a similar problem. Therefore, VRPSPD studies are first reviewed, and differences between this study and the study of Aydogdu and Ozyoruk (2020) are discussed later in this section.

VRPSPD studies began with the study of Min (1989), "The Multiple Vehicle Routing Problem with Simultaneous Delivery and Pickup Points." He found optimum routes for vehicles that delivered library materials from a main library to 22 branches and picked up the materials from branches to carry to the main library. Hezer and Kara (2013) developed a metaheuristic

method, bacterial foraging optimization algorithm, for VRPSPD. The proposed algorithm ensured superior results for 24 out of 40 test sets compared to the heuristic solution method developed by Dethloff (2011). Nagy and Salhi (2015) developed a heuristic solution method to be able to solve both VRPSPD and VRP with Mixed Pickup and Delivery (VRPMPD). They also extended the solution method to solve the multi-depot VRPSPD. Dell'Amico et al. (2006) proposed a branch and price algorithm to solve VRPSPD for 40 requests. Wassan et al. (2007) developed a reactive tabu search metaheuristic that could control feasibility of predetermined moves quickly and acted on iterations to guide the search. Ai and Kachitvichyanukul (2009) presented a particle swarm optimization method. Subramanian et al. (2010) presented a parallel algorithm based on the sequential heuristics. They embedded the algorithm with a multi-start heuristic containing the VND integrated to an iterated local search framework. Avci and Topaloglu (2015) proposed an adaptive local search algorithm for solutions of both VRPSPD and VRPMPD. Li et al. (2015) considered Multi-Depot VRPSPD and developed an iterated local embedded adaptive neighbourhood selection approach. Results showed that the proposed algorithm was superior than a large neighbourhood search, particle swarm optimization, and an ant colony optimization approach. Zachariadis et al. (2016) considered two-dimensional loading constraints beside simultaneous linehauls and backhauls. Kalayci and Kaya (2016) developed an ant colony system empowered variable neighbourhood search algorithm, a hybrid algorithm that combines an ant colony system and a variable neighbourhood approach. Experiments showed that the proposed algorithm provided better results compared to the individual performance of two methods. Kececi et al. (2015) considered a heterogeneous vehicle fleet consisting of different type of vehicles with costs. They proposed a mathematical model and an insertion-based heuristic for the solution. Yazgan and Buyukyilmaz (2017) developed a greedy heuristic solution method and carried out a regression analysis to find the relationship between travel distances and the number of customers and the capacity of vehicles. Goksal et al. (2013) presented a discrete particle swarm optimization method and a variable neighbourhood descent algorithm for VRPSPD. The algorithm kept swarm diversity by improving randomly selected solutions in each iteration. Montane and Galvao (2006) modified a tabu search algorithm to become compatible with VRPSPD. They developed three types of movements to observe inter-route adjacent solutions: the relocation, interchange, and crossover movements. Table 1 shows publications in terms of problem types, years, and solution methods. The review study of Koc and Laporte (2018) in those VRPSPD studies were classified in terms of models, solution methods, variants, and industrial applications, and case studies are advised to interested readers.

Table 1

Publications in terms of problem types and solution method

Year	Author	Problem	Solution Methods
1989	Min	Book Delivery and Pickup	Clustered Based Heuristic Method
2001	Dethloff	VRPSPD	Mathematical Model and Heuristic Method
2005	Nagy and Salhi	VRPSPD, Multi-Depot VRPSPD	Mathematical Model and Insertion-based Heuristic
2006	Montane and Galvao	VRPSPD	Tabu Search Algorithm
2006	Dell'Amico et al.	VRPSPD	Branch and Price Algorithm
2007	Wassan et al.	VRPSPD	Reactive Tabu Search Algorithm
2009	Ai and Kachitvichyanukul	VRPSPD	Particle Swarm Optimization Algorithm
2010	Subramanian et al.	VRPSPD	Parallel Algorithm Based on the Sequential Heuristics
2013	Hezer and Kara	VRPSPD	Bacterial Foraging Optimization Algorithm
2013	Goksal et al.	VRPSPD	Discrete Particle Swarm Optimization
2015	Avci and Topaloglu	VRPSPD and VRPMPD	Adaptive Local Search Algorithm
2015	Li et al.	Multi-Depot VRPSPD	Adaptive Neighbourhood Selection
2015	Kececi et al.	VRPSPD with heterogeneous vehicle fleet	Mathematical Model and Insertion-based Heuristic
2015	Nagy and Salhi	VRPSPD	Mathematical Model and Variable Neighbourhood Search
2016	Zachariadis et al.	VRPSPD with two-dimensional loading constraints	Local Search Algorithm
2016	Kalayci and Kaya	VRPSPD	Ant Colony Optimization
2017	Yazgan and Buyukyilmaz	VRPSPD	Insertion-based Heuristic
2020	Demirbilek	VRPSPD	Mathematical Model
2020	Aydogdu and Ozyoruk	DVRPSPD	Random Iterative Local Search Variable Neighbourhood Descending

Aydogdu and Ozyoruk (2020) considered two time periods for VRPSPD. Furthermore, they covered dynamic demands, 10% to 50% of total daily demands. They developed the Random Iterative Local Search Variable Neighbourhood Descending algorithm as a solution methodology and compared results to results of the mathematical model. Their aim was to minimize total travel times. In our study, 2 to 8 time periods and 50% to 87.5% dynamism levels are considered. The relationships between travel and waiting times based on the different lengths of routing periods and demand fluctuations are also examined. On one hand, shorter planning horizons can provide shorter waiting times for customers. On the other hand, longer ones ensure optimal routes as well as shorter travel times for vehicles and companies. Therefore, understanding the relationship between waiting and travel times under different routing period lengths helps companies to decide the best period length that establishes a trade-off between customer satisfaction and profit.

3. MATERIALS AND METHODS

In this section, the mathematical model for VRPSPD is first introduced. Next, the NNA is explained in detail.

3.1. Mathematical Model for VRPSPD

There are n customers that must be visited by k homogeneous vehicles. Each customer has a certain amount of pickup and delivery. When a vehicle visits a customer, the amount of pickup and delivery demands of a customer cannot exceed capacity of the vehicle. Vehicles must start from and return to the depot. The mathematical formulation for VRPSPD is mostly taken from the study of Montane and Galvao (2006). However, the maximum distance constraint is removed, and the maximum daily working time for vehicles and service times for customers are added into the model. The following shows notations and decision variables:

Notations:

V : set of customers

V_0 : set of customers and the depot

K : set of vehicles

c_{ij} : distance between locations of customers i and j

p_j : pickup demand of customer j , $\forall j \in V$

d_j : delivery demand of customer j , $\forall j \in V$

Q : vehicle capacity

T : working time limit for vehicles

S : service time

Decision Variables:

$x_{ijk} = 1$, if vehicle k moves from customer i to j . 0, otherwise.

y_{ij} = Cumulative amount of pickup at node i and transported in arc (i, j)

z_{ij} = Cumulative amount of delivery at node i and transported in arc (i, j)

The corresponding mathematical formulation is given by

Minimize:

$$\sum_k^K \sum_i^{V_0} \sum_j^{V_0} c_{ij} x_{ijk} + \sum_k^K \sum_i^{V_0} \sum_j^{V_0} S x_{ijk} \quad i \neq j \quad (1)$$

Subject to:

$$\sum_k^K \sum_i^{V_0} x_{ijk} = 1, \quad \forall j \in V \quad (2)$$

$$\sum_i^{V_0} x_{ijk} - \sum_i^{V_0} x_{jik} = 0, \quad \forall j \in V_0, \forall k \in K, i \neq j \quad (3)$$

$$\sum_j^V x_{0jk} \leq 1, \quad \forall k \in K \quad (4)$$

$$\sum_i^{V_0} y_{ij} - \sum_i^{V_0} y_{ji} = p_j, \quad \forall j \in V \quad (5)$$

$$\sum_i^{V_0} z_{ij} - \sum_i^{V_0} z_{ji} = d_j, \quad \forall j \in V \quad (6)$$

$$y_{ij} + z_{ij} \leq Q \sum_k^K x_{ijk}, \quad \forall i, \forall j \in V \quad (7)$$

$$\sum_i^{V_0} \sum_j^{V_0} c_{ij} x_{ijk} + S \sum_i^{V_0} \sum_j^{V_0} x_{ijk} \leq T, \quad \forall k \in K, i \neq j \quad (8)$$

$$x_{ijk} \in \{0,1\}, \quad \forall i, \forall j \in V_0, \forall k \in K, i \neq j \quad (9)$$

$$y_{ij}, z_{ij} \geq 0, \quad \forall i, \forall j \in V_0, i \neq j \quad (10)$$

The objective function (1) is to minimize total travel and service times. Constraints (2) make sure that each customer is visited by exactly one vehicle. Constraints (3) ensure that the same vehicle arrives and departs from each customer. Constraints (4) show the maximum number of vehicles allowed. Constraints (5) and (6) are flow equations for pick-up and delivery, respectively; they guarantee that both are satisfied for each customer. Constraints (7) ensure that pickup and delivery demands do not exceed capacity of the vehicle moving from customer i to j . Constraints (8) show travelling and service times for a vehicle do not exceed the allowed working time. Constraints (9) and (10) demonstrate boundaries of decision variables.

3.2. The Nearest Neighbourhood Algorithm (NNA)

The NNA is one of the most popular heuristic solution methods to solve VRPs and variants in the literature. The main idea behind it is to find the nearest neighbour in terms of the distance to a reference point (customer or depot) and insert it to the schedule. This procedure is repeated until all customers are assigned or the schedule is no longer available. Since we deal with VRPSPD, the pickup and delivery demands of customers as well as capacity levels of vehicles as constraints must be considered. Therefore, when assigning customers in terms of seeking minimum distances, the demands of customers and

the capacities of vehicles are taken into consideration. If the delivery and pickup demands of a customer violates the capacity of a vehicle, the algorithm does not assign it to the schedule even though it has the minimum distance cost.

The other important factor is the waiting time. The NNA is originally designed to consider distances between customers when assignment decisions are made. However, waiting times for customers must be considered since the arrival of customers in different time periods causes significant waiting times. Let us assume that there are 3 vehicles, V_1 , V_2 , and V_3 , customers are visited by these three vehicles as illustrated in Fig. 1. In this example, two time periods exist, at morning and noon, and the rerouting activity must be made at noon. The last customers are visited at 11:00, 11:20, and 10:50 am by V_1 , V_2 , and V_3 , respectively. If the earliest time the first customer (red one) can be visited is 12:00, the algorithm checks waiting times, basically the differences between the finish times of the previous services and the earliest starting times of the next customers. It is assumed that waiting times, w_1 , w_2 , and w_3 , are 60, 40, and 70 minutes for V_1 , V_2 , and V_3 respectively and travel times between the last customers visited, and the first customer after rerouting, d_1 , d_2 , and d_3 , are 20, 35, and 10 minutes. The algorithm selects V_3 to visit the first customer if minimizing distances is considered as the objective function. However, the vehicle must wait 60 minutes to be able to start the service. If V_2 is selected to visit the customer, the vehicle spends 40 minutes, 35 minutes during traveling and 5 minutes during waiting. Note that we refer the cost of the travelling and waiting of vehicles as “average travel time per visit” in the remaining part of the study since fuel consumption of vehicles are ignored.

It is important to note that the NNA works for each time interval as a static fashion. In other words, the algorithm only considers new and unvisited customers when routes are constructed at the beginning of an associated period. Customers arriving after routes are constructed must wait till the next period. Therefore, each period in a day is actually a static VRPSDP. However, considering many periods and inserting customers arriving during the planning horizon to the daily routes adds a degree of dynamism to the problem.

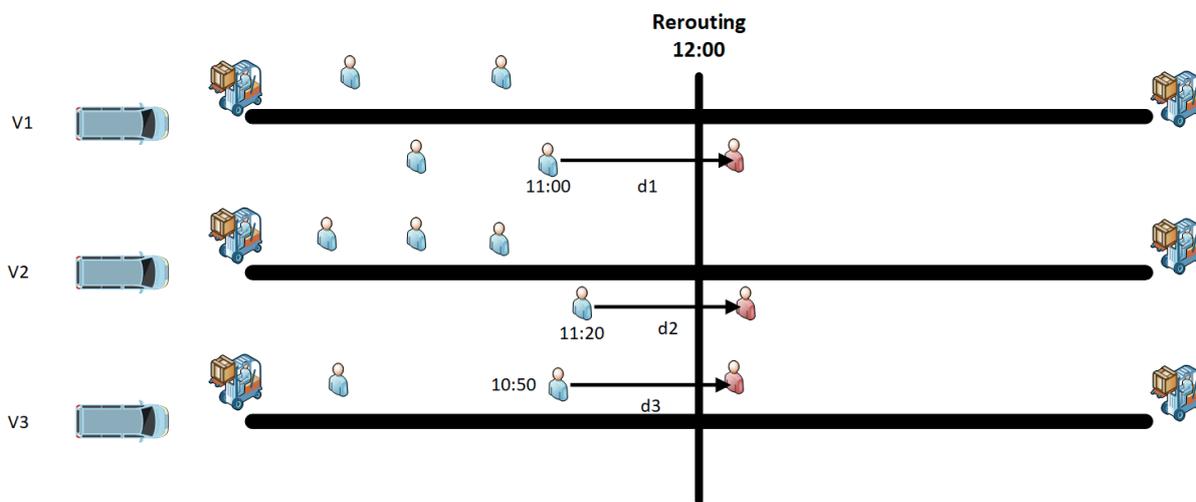


Figure 1. Illustration of the rerouting activity and the decision of customer visits

Fig. 2 shows the pseudocode for the NNA. First, the number of routing periods and vehicles are defined. In each period, a static time interval, new and unvisited customers are added to the “Customers” list on Line 4. Between Line 5 and 25, NNA works for the defined customers and vehicles. In each iteration, the cheapest insertion cost is calculated for each of the customers by considering all vehicles. After insertion costs are calculated for each customer as between Line 7 and 22, the customer with the cheapest insertion cost is assigned to the vehicle and removed from the list if the capacity and demand constraints are not violated. This procedure continues till all customers are scheduled.

4. COMPUTATIONAL EXPERIMENTS AND RESULTS

Three time periods, 60, 120, and 240 minutes, are considered to be able to understand the effect of period length on waiting and travel times for same customer sets. Since the daily working time of vehicles is defined as 480 minutes, there are 8, 4, and 2 routing periods that new customers can arrive and rerouting activities can be made. The expected number of customers

for a day is divided by the number of periods to be able to calculate the number of customers scheduled for the current time period. Let us assume that a 60-minute period length (8 periods) is considered and 48 customers for that day are expected. Thus, the number of customers that arrived in each period is 6. For a 120-minute time period (4 periods), the arrival of 12 customers will be expected. In real life, it is hard to observe that same number of customers arriving to the system in each time period, of course. By considering this possibility, period by period, demand fluctuations are also inspected. We randomly increase or decrease the number of customers by 30% from the expected values. Based on the above example, between 8 and 16 customers in each 120-minute routing period are expected.

Algorithm 1 The Nearest Neighbourhood Algorithm

```

1: Initiate Periods (P)
2: Set Vehicles (V)
3: for p= 1 To P do
4:   Set New and Unvisited Customers (C)
5:   while Customers list is not empty do
6:     MinCustomerCost ← ∞
7:     for c= 1 To C do
8:       MinVehicleCost ← ∞
9:       for v= 1 To V do
10:        Calculate insertioncost of Customers[c] for Vehicles[v]
11:        if insertionCost =< MinVehicleCost then
12:          if c satisfies capacity and time constraints for Vehi-
13:          cles[v] then
14:            MinVehicleCost←insertionCost
15:            Index←v
16:          end if
17:        end if
18:      end for
19:      if MinVehicleCost =< MinCustomerCost then
20:        MinCustomerCost←MinVehicleCost
21:        CustomerIndex←c
22:      end if
23:    end for
24:    Vehicles [Index]← Customers[CustomerIndex]
25:    Remove Customers[CustomerIndex] from Customers list
26:  end while
27:  Move Next Period
28: end for
29: Print Waiting and Travel Times

```

Figure 2. Pseudocode for NNA modified for DVRPSPD

There are some methods that can be used to measure the degree of dynamism (DD) in the literature. The method of Lund et al. (1996) is one of the most common methods. In this method, the DD is calculated by dividing the number of dynamically arriving customers by the total number of customers. In this study, dynamism levels change according to the length of periods since the algorithm starts with the number of customers arriving before the first period. Therefore, the first period in each day begins with known customers, and the remaining periods contain dynamic customers. In this situation, the DD level increases with the number of time periods accordingly. For example, if the 120-minute period length is considered, there are four routing periods in the 480-minute day length. The first period is composed of known customers. Thus, the remaining three contain dynamic customers, and the DD will be 0.75 (3/4). In this condition, there are three DD levels, 0.875, 0.75, and 0.5 for 60, 120, and 240-minute period lengths. Note that the case that all customers are known at the beginning is also considered for the comparison purpose.

As it is mentioned before, the day length is 480 minutes. The service time for each customer is 5 minutes. To test both models only for one customer set does not provide accurate results since results can significantly change for another customer set. Therefore, 30 sets that include different customers' data are taken into consideration. For each test, locations of customers and pickup and delivery demands are changed. It can be considered as operations of a company for 30 days. Three customer sets, 48, 96, and 144, are considered. Delivery demands of customers are assigned uniformly between 4 and 10 packages, and pickup demands are assigned uniformly between 3 and 7 packages. There is a different number of vehicles depending on the number of customers. Each vehicle capacity is 75 packages. A sufficient number of vehicles, as all customers can be visited by satisfying their pickup and delivery demands, is defined. The total area size is 3600 km² (60 km X 60 km) and represented, $X \in [0,60]$ and $Y \in [0,60]$, in the coordinate system. The depot is located at the middle of the service area. All related data are shown in Table 2. Finally, NNA is coded with Python Programming Language, and all tests are conducted on a PC with an Intel i5 7200U 2.5 GHz CPU and 8 GB Ram.

Table 2

Test settings

Daily working time (minute)	480
Service time (minute)	5
Service area	$X \in [0,60]$ and $Y \in [0,60]$
Number of customers	48,96,144
Demand fluctuation	$\pm 30\%$
Vehicle capacity (package)	75
Pickup demand (package)	Uniform (3,7)
Delivery demand (package)	Uniform (4,10)
Time period (minute)	60,120,240
Degree of dynamism	0.875,0.75,0.5

Table 3 shows travel times (minute) per visit and waiting times (minute) per customer under the stable demand structure. Confidence intervals with a 95% confidence level are also provided. For example, an average travel time per visit can be between 33.85 and 34.95 minutes (34.40 ± 0.55) with 95% confidence level if the 48 customers and 60-minute routing period test set are considered. First, average travel times per visit decrease when the length of periods increases as expected. A 480-minute period demonstrates the case that all customers are known at the beginning. Compared to the 60-minute time period (87.5% DD), average travel times per visit decrease by the least, more than 50%, if all customers are known before optimizations start. This is called "Value of Information" and shows that the difference between fully static and dynamic solutions for the same problem set. When the number of customers increases from 48 to 144, average travel times per visit slightly decrease. However, once we know all customers at the beginning, average travel times per visit decrease by 11% and 18% when increasing the number of customers to 96 and 144. In this study, waiting times for customers that represent differences (minute) between arrival times and starting times to services are also taken into consideration. Waiting times for a fully static case (480-minute day length) are not calculated since we should know when each customer arrives on the previous day for accurate calculations. This is a limitation for this study and discussed in the Conclusion Section. Although average travel times per visit change between 9% and 23% period by period, waiting time sensitivity based on different time periods is relatively enormous. If the period length increases 60 to 120, each customer must wait almost 62 minutes more to

be serviced whereas vehicles spend 7 minutes less per visit during travelling. When the number of customers increases, waiting times per customer increase whereas average travel times stay almost the same.

Table 3

*Average travel times (minute) per visit and waiting times (minute) per customer under the stable demand structure (*CI: Confidence Interval +/-)*

Customers	Time Periods	Travel Time (TT)	*CI for TT (\pm)	Waiting Time (WT)	*CI for WT (\pm)
48	60	34,40	0,55	4,63	0,53
	120	31,15	0,80	34,07	1,67
	240	27,84	0,60	66,39	2,60
	480	16,67	0,34
96	60	34,31	0,55	5,42	0,39
	120	28,89	0,35	41,05	1,03
	240	26,57	0,41	76,83	1,36
	480	14,81	0,21
144	60	33,47	0,44	6,79	0,32
	120	28,41	0,32	41,07	0,82
	240	25,59	0,31	78,73	1,05
	480	13,63	0,12

Table 4 shows average travel times (minute) per visit and waiting times (minute) per customer under demand fluctuations. As mentioned, the number of customers arriving in each period changes $\pm 30\%$ based on expected values. First, average travel times per visit increase compared to the stable demand scenario. On the other hand, average waiting times per customer are slightly less than average waiting times in the stable demand scenario. Similarly, average waiting times are very sensitive to time periods. When the length of time periods increases, waiting times per customer sharply escalate. Overall, results under demand fluctuations follow the same pattern with results under the stable demand condition. Although it sometimes seems there are only several minute differences per customer between two results, it has a significant effect when considering all customers. For example, the difference between the average travel time of a 60-minute period and a 240-minute period is around 8 minutes per visit. If we consider 144 customers/visits, it costs 1152 minutes daily. It means that vehicles spend more than 19 hours during daily travels if the 60-minute routing period is applied.

Table 4

*Average travel times (minute) per visit and waiting times (minute) per customer under demand fluctuations (*CI: Confidence Interval +/-)*

Customers	Time Periods	Travel Time (TT)	*CI for TT (\pm)	Waiting Time (WT)	*CI for WT (\pm)
48	60	38,99	0,76	3,67	0,37
	120	33,76	0,72	32,00	1,27
	240	29,60	0,95	68,40	2,55
	480	16,67	0,34
96	60	35,65	0,44	5,73	0,56
	120	31,27	0,54	35,41	1,24
	240	27,34	0,85	74,77	2,61
	480	14,81	0,21
144	60	35,65	0,50	6,23	0,46
	120	29,72	0,45	40,34	1,12
	240	27,23	0,74	82,07	2,29
	480	13,63	0,12

5. CONCLUSION

Vehicle Routing Problem with Simultaneous Pickup and Delivery is getting more attentions day by day since optimal or near optimal solutions significantly decrease operational costs of companies by delivering and collecting goods at the same time. Although many studies that consider static problem settings where all customer information is known at the beginning exist in the literature, the dynamism issue where customers arrive during a planning period and schedules are evolving dynamically is rarely taken into account. In this study, we consider the dynamism issue as static routing periods when optimizations are made. In other words, we find optimal routes for new customers and unvisited customers assigned from previous periods in each routing period. The Nearest Neighbourhood Algorithm is used to solve optimization problems quickly. Some real-life

aspects such as different numbers of customers, daily working times for vehicles, service times for customers, and demand fluctuations for routing periods are considered as parameters and constraints. Moreover, different degree of dynamism (DD) levels, 50% to 87.5%, are examined to observe how different DD levels affect results. Since results can significantly change based on divergent data sets, 30 different data sets that include a variety of customers' locations and pickup and delivery demands are evaluated, and confidence intervals are calculated for average travel times per visit and waiting times per customer to provide more accurate results for readers.

First, as expected, considering dynamic customer arrivals significantly increases average travel times per visit. On the other hand, when the length of periods decreases, the average waiting times also sharply decrease. If routes are constructed 8 times a day, each customer waits an average 5 minutes compared to 66 minutes if it is made 4 times a day. In this situation, each vehicle must spend seven more minutes visiting each customer. When the demand fluctuation case is examined, longer average travel times are observed. Furthermore, average travel times per visit decrease, and waiting times per customer increase slightly when the number of customers goes up. Note that several minute differences among results per customer/visit make a huge impact when considering the high number of customers. On one hand, companies try to avoid unnecessary travel times to reduce operational costs. On the other hand, customers demand to be served as soon as possible. In this study, we attempt to show the relationship between travel times and waiting times to help decision makers find the best solution for both companies and customers.

There are some limitations and assumptions for this study as well. First, when calculating waiting times, it is assumed that customers in the first period are ready to be serviced. However, these customers are supposed to arrive the previous day, so waiting times can be higher than those calculated here. Although this can be achieved by defining interarrival times for customers, we have a different number of customers visited scenario by scenario in this time, and it causes miscalculations and miscomparisons for travel and waiting times. Furthermore, pickup and delivery demands and the capacity of vehicles are considered the same, test by test. Testing a variety of demands and capacity levels can also be useful. However, the first aim of this study is to examine how the length of periods affect waiting and travel times per customer/visit.

In future research, each customer can be considered as a decision point. Instead of considering a specific time period, whenever a new customer arrives to system, the routing and scheduling activity is made to find the optimum visit time for the customer.

Peer-review: Externally peer-reviewed.

Conflict of Interest: The authors has no conflict of interest to declare.

Grant Support: The author declared that this study has received no financial support.

Hakem Değerlendirmesi: Dış bağımsız.

Çıkar Çatışması: Yazar çıkar çatışması bildirmemiştir.

Finansal Destek: Yazar bu çalışma için finansal destek almadığını beyan etmiştir.

References/Kaynaklar

- Ai, T. J., & Kachitvichyanukul, V. (2009). A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, 36(5), 1693-1702.
- Avcı, M., & Topaloglu, S. (2015). An adaptive local search algorithm for vehicle routing problem with simultaneous and mixed pickups and deliveries. *Computers & Industrial Engineering*, 83, 15-29.
- Aydoğdu, B., & Özyörük, B. (2020). Dinamik eş zamanlı topla dağıt araç rotalama probleminin çözümü için matematiksel model ve sezgisel yaklaşım: Rassal iteratif yerel arama değişken komşu iniş algoritması. *Journal of the Faculty of Engineering & Architecture of Gazi University*, 35(2).
- B. Columbus, "Book Reviews," no. September, pp. 558–559, 1972.
- Chen, J. F., & Wu, T. H. (2006). Vehicle routing problem with simultaneous deliveries and pickups. *Journal of the Operational Research Society*, 57(5), 579-587.
- Christofides, N., & Eilon, S. (2006). An algorithm for the vehicle-dispatching problem. *Journal of the Operational Research Society*, 20(3), 309-318.
- Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management science*, 6(1), 80-91.
- Dell'Amico, M., Righini, G., & Salani, M. (2006). A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection. *Transportation science*, 40(2), 235-247.

- Demirbilek, M. (2020). A Tactical/Strategic Level Cost Analysis Based On Visit Time Preferences For Vehicle Routing Problem With Simultaneous Pickup And Delivery. *European Journal of Technique (EJT)*, 10 (2), 301-312.
- Dethloff, J. (2001). Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up. *OR-Spektrum*, 23(1), 79-96.
- Goksal, F. P., Karaoglan, I., & Altıparmak, F. (2013). A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery. *Computers & Industrial Engineering*, 65(1), 39-53.
- Hezer, S. & Kara, Y. (2013). Eşzamanlı dağıtım ve toplama araç rotalama problemlerinin çözümü için bakteriyel besin arama optimizasyonu tabanlı bir algoritma. *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 28(2).
- Kalaycı, C. B., & Kaya, C. (2016). An ant colony system empowered variable neighbourhood search algorithm for the vehicle routing problem with simultaneous pickup and delivery. *Expert Systems with Applications*, 66, 163-175.
- Kececi, B., Altıparmak, F., & Kara, I. (2015). Heterogeneous vehicle routing problem with simultaneous pickup and delivery: Mathematical formulations and a heuristic algorithm. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 30(2), 185-195.
- Koç, Ç., & Laporte, G. (2018). Vehicle routing with backhauls: Review and research perspectives. *Computers & Operations Research*, 91, 79-91.
- Laporte, G. (2009). Fifty years of vehicle routing. *Transportation science*, 43(4), 408-416.
- Li, J., Pardalos, P. M., Sun, H., Pei, J., & Zhang, Y. (2015). Iterated local search embedded adaptive neighbourhood selection approach for the multi-depot vehicle routing problem with simultaneous deliveries and pickups. *Expert Systems with Applications*, 42(7), 3551-3561.
- Lund, K., Madsen, O. B., & Rygaard, J. M. (1996). Vehicle routing problems with varying degrees of dynamism. IMM Institute of Mathematical Modelling.
- Min, H. (1989). The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research Part A: General*, 23(5), 377-386.
- Montané, F. A. T., & Galvao, R. D. (2006). A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers & Operations Research*, 33(3), 595-619.
- Nagy, G., & Salhi, S. (2005). Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *European journal of operational research*, 162(1), 126-141.
- Subramanian, A., Drummond, L. M. D. A., Bentes, C., Ochi, L. S., & Farias, R. (2010). A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, 37(11), 1899-1911.
- Wassan, N. A., & Nagy, G. (2014). Vehicle routing problem with deliveries and pickups: modelling issues and meta-heuristics solution approaches. *International Journal of Transportation*, 2(1), 95-110.
- Yazgan, H. R., & Büyükyılmaz, R. G. (2017). Eş zamanlı topla dağıt araç rotalama problemine sezgisel bir çözüm yaklaşımı. *Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 22(2), 436-449.
- Zachariadis, E. E., Tarantilis, C. D., & Kiranoudis, C. T. (2016). The vehicle routing problem with simultaneous pick-ups and deliveries and two-dimensional loading constraints. *European Journal of Operational Research*, 251(2), 369-386.

