processes

Article

Vehicle Routing Problem Model with Practicality

SeJoon Park [1], Chunghun Ha [2] and Hyesung Seok [2,*]

[1] Division of Energy Resources Engineering and Industrial Engineering, Kangwon National University, KNU Chuncheon Campus, 1 Gangwondaehakgil, Chuncheon-si 24341, Gangwon-do, Republic of Korea
[2] Department of Industrial & Data Engineering, Hongik University, 94 Wausan-ro, Mapo-gu, Seoul 04066, Republic of Korea
* Correspondence: hseok@hongik.ac.kr; Tel.: +82-2-320-1658

Abstract: Truck platooning has recently become an essential issue in automatic driving. Though truck platooning can increase safety and reduce fuel consumption and carbon emissions, the practical vehicle routing problem involved in truck platooning has not been sufficiently addressed. Therefore, we design a mixed-integer linear programming model for the routing problem in truck platooning considering the deadline of vehicles, continuous-time units, different fuel reduction rates, traffic congestion avoidance, and heterogeneous vehicles. In addition, a forward–backward heuristic called the "greedy heuristic" is presented for reasonable computation time. To validate the model's performance, several parameters, such as the percentage of fuel reduction, percentage of detour vehicles, and percentage of platooned links (road segments), are considered. Additionally, various cases are considered with varying fuel reduction rates, traffic flow rates, and time windows.

Keywords: automatic driving; cost minimization; forward–backward heuristics; greedy algorithm

Citation: Park, S.; Ha, C.; Seok, H. Vehicle Routing Problem Model with Practicality. *Processes* **2023**, *11*, 654. https://doi.org/10.3390/pr11030654

Academic Editor: Chien-Chih Wang

Received: 27 December 2022
Revised: 14 February 2023
Accepted: 15 February 2023
Published: 21 February 2023

1. Introduction

Vehicle platooning is an autonomous-driving-based method, according to which vehicles are driven in a group with a short distance being maintained simultaneously between them. Because this driving method enables a reduction in the air resistance to subsequent vehicles, the fuel consumption and the air pollution of the vehicles can be reduced. Additionally, platooning reduces the risk of accidents and is effective for increasing road capacity and utilization as well as reducing traffic congestion. Currently, major countries around the world vie for leadership in research on vehicle platooning. In Europe and the United States, the road test phase is already underway, following the completion of the theoretical research phase. In Europe, the fuel-saving effect was verified through a road test involving truck platooning [1]. In the United States, vehicle platooning was applied to eight passenger cars on a highway to verify its effectiveness [2].

To employ vehicle platooning commercially on public roads, not only implementation technologies but also numerous operational preparations, such as institutional establishments, road infrastructure construction, and cost-effective operation methods, are required. Therefore, in this study, the problem regarding the effective operation in vehicle platooning was addressed, especially for more practical applications.

For several decades, various vehicle routing problems (VRPs) have been studied, considering varying numbers and types of nodes, the number and characteristics of the vehicles, and customer conditions. However, the VRP for platooning (VRPP) is fundamentally different from the standard VRP and its variants in the following ways. First, in the VRPP, multiple vehicles have independent origins and destinations. Second, the route and schedule of each vehicle are independent, but they become dependent on platooning. Third, vehicles are allowed to wait and detour during platooning. Fourth, there is a time window (available starting time and maximum allowable arrival time) considered for the vehicles. In brief, the VRPP can be defined as the problem regarding determining the optimal route and schedule for vehicles that are allowed to wait and detour during platooning to minimize the costs in the cases of vehicles having their own origins, destinations, and time windows.

To address the optimization of the VRPP, the VRPP is defined using a mathematical programming model. Its optimization has been investigated in several studies [3,4]. However, existing VRPP models present drawbacks; for example, they do not consider the deadline of vehicles [3] or they use a discrete-time unit [4]. The model by Larson et al. (2016) is quite similar to ours, but they used some different variables and constraints [5]. Our model can also be employed when the fuel reduction rates of the leading and following vehicles are different, which is more realistic. Constraints to avoid high congestion are also designed in our model. Moreover, the computation time is more than 100 s in most cases of their example. We propose a novel greedy heuristic model in which computation time is only a few seconds. A more detailed comparison with previous studies will be described in Section 2.

In brief, the contribution of our study is to propose a mathematical programming model to overcome the disadvantages of the existing VRP by considering the following.

(1) The deadline for vehicles is considered;
(2) Continuous-time unit is applied for reality;
(3) The different fuel reduction rates of the leading and following vehicles are employed;
(4) The constraints to avoid traffic congestion are employed;
(5) A new greedy heuristic model is designed for more practical use with less computation time;
(6) The platooning among heterogeneous vehicles, where the dispatching and routing are centrally controlled, is considered.

The remainder of this paper is organized as follows. In Section 2, previous studies related to vehicle platooning are reviewed, and the direction of this study is presented. In Section 3, a mathematical programming model for the VRPP is presented. The proposed heuristic principles and pseudocode for the VRPP are presented in Section 4. In Section 5, an experimental environment based on Germany, Japan, and Korea and the results of the VRPP are presented. Additionally, the parameters that can affect the vehicle-platooning efficiency are analyzed, and their effects are verified via statistical methods. Finally, in Section 6, the study is summarized, and plans for future research are discussed.

2. Literature Review on Vehicle Platooning

2.1. Subsection Vehicle Routing Problem for Platooning (VRPP)

Automatic-driving technology enables vehicles to move autonomously. Vehicles can be classified as manual, adaptive cruise control (ACC), or cooperative adaptive cruise control (CACC) according to the type of automatic-driving technology [6]. In the case of ACC, a vehicle uses a sensor to analyze situations on its own and conducts automatic driving. It drives while calculating the distance from the vehicle in front of it. In the case of CACC, the entire vehicle flow can be identified during autonomous driving, via communication technology.

Segata et al. first developed a simulation tool for situations in which automated technologies, such as ACC and CACC, and nonautomated ones, such as vehicles driven by humans, coexist [7]. Then, a traffic assistant system for vehicle platooning of manual, ACC, and CACC vehicles was additionally investigated [6]. Various maneuvers, such as forming, adjusting, splitting, dismissing, and joining, have been achieved through simulations. The effects of the market penetration of CACC and the platoon size on freeway traffic have also been examined.

Because one of the main goals of vehicle platooning is to reduce pollution, many researchers considered the pollution-routing problem. Nasri et al. studied the pollution-routing problem regarding autonomous vehicles to reduce emissions, fuel consumption, and travel times under uncertain traffic conditions [8]. Dukkanci et al. modeled the pollution-routing problem combined with a location-routing problem [9]. Similarly, Demir et al. [10] investigated an adaptive large neighborhood search for the pollution-routing problem to determine the speed on each route segment to minimize fuel consumption, emissions, and driver costs. Additionally, a metaheuristic for a speed and departure-time optimization algorithm for the pollution-routing problem was proposed [11].

Table 1 compares our study to the recent research of the routing problem in vehicle platooning [12]. The studies without a speed constraint assume a fixed driving speed.

Table 1. Comparison of our study to previous VRPP models.

	Objective		Constraints				Decisions				Dynamics		
No.	**F**	**D**	**T**	**S**	**D**	**L**	**PC**	**R**	**S**	**SC**	**SS**	**RT**	**O**
[3]	∨		∨				∨	∨	∨		∨		
[13]	∨		∨	∨			∨	∨		∨	∨	∨	
[14]	∨		∨		∨		∨	∨	∨		∨		∨
[5]	∨		∨		∨		∨	∨	∨		∨		
[4]	∨		∨				∨	∨	∨		∨		
[15]	∨		∨		∨		∨	∨	∨		∨		
This study	∨		∨		∨		∨	∨	∨		∨		

Objective: F (fuel), D (delays); constraints: T (timing), S (speed), D (detour), L (length); decisions: PC (platoon composition), R (route), S (schedule), SC (speed changes); dynamics: SS (schedule), RT (real-time), O (opportunistic).

Especially, Sokolov et al. (2017), Nourmohammadzadeh and Hartmann (2019), and Larson et al. (2016) have similar assumptions and considerations to those of our model [5,14,15]. However, Sokolov et al. (2017) do not allow vehicles to wait at intermediate nodes. In our model, vehicles can wait at any node if deadline requirements are satisfied [14]. The model by Nourmohammadzadeh and Hartmann (2019) is significantly different from ours because they use a discrete unit time slot, i.e., 5 min, 10 min, and so on [15]. We use continuous time in decision variables. Hence, in our model, vehicles can start from any node at any time if the constraints are satisfied. In detail, Nourmohammadzadeh and Hartmann set the schedule of each vehicle as to the binary variable in terms of arc between i and j, and time slot t. If the value of the binary variable is 1, it means the vehicle passes the arc at time *t*. Hence, the complexity of the problem is the product of the number of vehicles, the number of arcs, and the number of time slots. Therefore, to use the number of nodes/arcs in their study is not applicable to our model. Each arc in their study is just the time unit. In other words, the number of arcs between intersections in their study is regarded as 1 in our study because we use continuous time.

The mathematical model of Larson et al. (2016) is quite similar to ours [5]. However, they used some different variables, such as each vehicle's variable for platooning and the waiting time variable at each node. Using a one-time variable is enough in our model while Larson et al. (2016) used two time variables, including the waiting time variable. In addition, we build the constraints to avoid high congestion. One of the advantages of our model can consider the effect of leading vehicle fuel reduction, especially in case the fuel reduction of a leading vehicle and following vehicles are different.

Moreover, they modified the model in GAMS and add more constraints and rules to reduce the size of the computation. However, the computation time is more than 100 s with a 1% optimality gap in most cases. In our study, we propose a more practical greedy heuristic model in which computation time is only a few seconds.

2.2. VRPP and VRP Variants

To understand our model clearly, let us compare the VRP with time windows (VRPTW) and VRPP at first. The vehicle routing problem in platooning can be regarded as a VRPTW or platooning. The traditional VRPTW aims to cater to customers within predefined time windows at the minimum cost, satisfying the capacity and total travel time constraints of each vehicle [16]. This is generally an NP-hard (Nondeterministic polynomial-time hardness) problem. Therefore, various heuristics, such as the genetic algorithm, evolutionary algorithms, tabu search, and ant colony optimization algorithm, are used to solve the problem.

Also, in some studies, the parts of nodes are intentionally removed by pre-processing, e.g., Luo et al. (2018) [17]. In this study, a heuristic algorithm was also developed to address the VRPTW in the platooning model more effectively, and testing was performed on actual data. The differences between the traditional VRPTW and the VRPP are presented in Table 2. A strict time window and realistic constraints were applied, such as (1) equal speed and (2) different departures and destinations for each vehicle in the same group (Table 2).

Table 2. Comparison between VRPTW and VRPP.

	VRPTW	**VRPP**
Dependency of vehicles	Independent	Dependent
Heterogeneity of vehicles	N/A	Yes
Equal speed constraint	No	Yes
Type of problem	NP hard	NP hard
Number of departures/destinations	Single (or same)	Multiple (or different)
Tradeoff between time reduction and detouring	No	Yes

A comparison between other VRP variants and the VRPP is presented in Table 3. The Capacitated VRP (CVRP) considers the homogeneous depots and vehicles when there are no service time constraints. The heterogeneous fleet VRP or mixed fleet VRP (HFVRP) considers heterogeneous depots and vehicles without service time constraints. The VRP with time windows (VRPTW) considers the service time constraints. The VRP with pickup and delivery (VRPPD) deals with the pickup and delivery at a visit by heterogeneous vehicles. The VRP with backhauls (VRPB) considers the pickup and delivery by homogeneous vehicles. The multi-depot VRP (MDVRP) has heterogeneous types of depots. The periodic VRP (PVRP) deals with the multiple visits of vehicles. As indicated by Tables 2 and 3, the VRPP addressed in this study differs from the conventional VRP and its variants in the following ways.

Table 3. Comparison between VRP variants and VRPP.

<table>
<tr><th colspan="2" rowspan="2">Problem</th><th rowspan="2">Type of Depots</th><th colspan="4">Vehicles</th><th colspan="3">Customers</th><th>Reference</th></tr>
<tr><th>Set</th><th>Capacity</th><th>Vehicle Time Window</th><th>Same Schedule on the Link</th><th>Service</th><th>Vehicle Visit</th><th>Service Time Window</th><th></th></tr>
<tr><td rowspan="7">VRP</td><td>CVRP</td><td>S = D</td><td>HO</td><td>HO</td><td>$(0, \infty)$</td><td>Not required</td><td>ns</td><td>once</td><td>$(0, \infty)$</td><td>Laporte [18]</td></tr>
<tr><td>HFVRP</td><td>S = D</td><td>HE</td><td>HE</td><td>$(0, \infty)$</td><td>Not required</td><td>ns</td><td>once</td><td>$(0, \infty)$</td><td>Leung et al. [19]</td></tr>
<tr><td>VRPTW</td><td>S = D</td><td>HO</td><td>HO</td><td>$(0, \infty)$</td><td>Not required</td><td>ns</td><td>once</td><td>$\{t_s, t_c\}$</td><td>Bräysy and Gendreau [20]; Gendreau and Tarantilis [21]</td></tr>
<tr><td>VRPPD</td><td>S = D</td><td>HE</td><td>HE</td><td>$(0, \infty)$</td><td>Not required</td><td>$p, d, p\&d$</td><td>once</td><td>$(0, \infty)$</td><td>Tasan and Gen [22]</td></tr>
<tr><td>VRPB</td><td>S = D</td><td>HO</td><td>HO</td><td>$(0, \infty)$</td><td>Not required</td><td>p, d</td><td>once</td><td>$(0, \infty)$</td><td>Pradenas et al. [23]; Brar and Saini [24]</td></tr>
<tr><td>MDVRP</td><td>S = D</td><td>HO</td><td>HO</td><td>$(0, \infty)$</td><td>Not required</td><td>ns</td><td>once</td><td>$(0, \infty)$</td><td>Montoya-Torres et al. [25]</td></tr>
<tr><td>PVRP</td><td>S = D</td><td>HO</td><td>HO</td><td>$(0, \infty)$</td><td>Not required</td><td>ns</td><td>multiple</td><td>$(0, \infty)$</td><td>Campbell and Wilson [26]; Gulczynski et al. [27]</td></tr>
<tr><td colspan="2">VRPP</td><td>$S \neq D$</td><td>HE</td><td>HE</td><td>$\{t_s, t_c\}$</td><td>Required for each group</td><td>-</td><td>-</td><td>-</td><td>This article</td></tr>
</table>

Capacitated VRP (CVRP), heterogeneous fleet VRP or mixed fleet VRP (HFVRP), VRP with time windows (VRPTW), VRP with pickup and delivery (VRPPD), VRP with backhauls (VRPB), multi-depot VRP (MDVRP), periodic VRP (PVRP). *HO*: homogeneous, *HE*: heterogeneous, *S*: a set of starting depot locations, *D*: a set of destination depot locations, t_s: starting time, t_c: completion time, *ns*: not specified, *p*: pickup only, *d*: delivery only, *p&d*: pickup and delivery at a visit.

First, in VRPs, each vehicle is independent; however, in the VRPP, the vehicles are interdependent. The VRP assumes that each vehicle drives independently, although the optimal paths of the vehicles are linked to each other. Therefore, the time adjustment and detour of the vehicle are only related to the feasibility. However, the VRPP becomes more cost-effective as the group size increases; thus, the driving times and routes of all the vehicles must be controlled at all times. That is, in the VRPP, the time window and detouring are fundamental conditions, which makes the problem more difficult. This characteristic is described in the column of the same schedule on the link.

Second, the VRP generally considers one origin and one destination. However, the VRPP generally considers multiple origins and destinations.

Third, because the speed limit of the passenger vehicle is higher than the speed limit of the truck on most roads, vehicles of different sizes are difficult to drive in a group except in special cases. The group can be aggregated and segregated frequently, and vehicles belonging to a group in one section are likely to have different origins, destinations, starting times, and arrival times. Thus, the VRPP can be defined as the problem regarding optimizing the route in consideration of time adjustment and detouring so that vehicles with different origins, destinations, and starting and arrival times can move autonomously in a group (of as many vehicles as possible) in road sections. At this time, there is a tradeoff relationship between the reduction in the fuel costs owing to group driving and the increase in the costs of owing to time adjustment and detouring.

3. Model

3.1. Problem Description

There are two main cases in which vehicle platooning can be employed. In case 1, the platooning involves adjustment of the schedules of vehicles crossing the same route. In case 2, the routes of vehicles crossing different routes are changed so that all or parts of the routes of the vehicle are common. Figure 1 shows a network with a city as a node and a road segment as a link. In case 1, for example, if there are two vehicles traveling from city 1 to city 4, as per Figure 1, and they leave city 1 at the same time, the fuel-conserving effect can be obtained via platooning. In the VRPP problem of this study, the origin and destination of the vehicle are predetermined, along with the earliest departure time at the origin and the latest arrival time at the destination. These two times are defined as the time schedule of each vehicle. In case 1, whether platooning should be performed can be determined according to the time schedule.

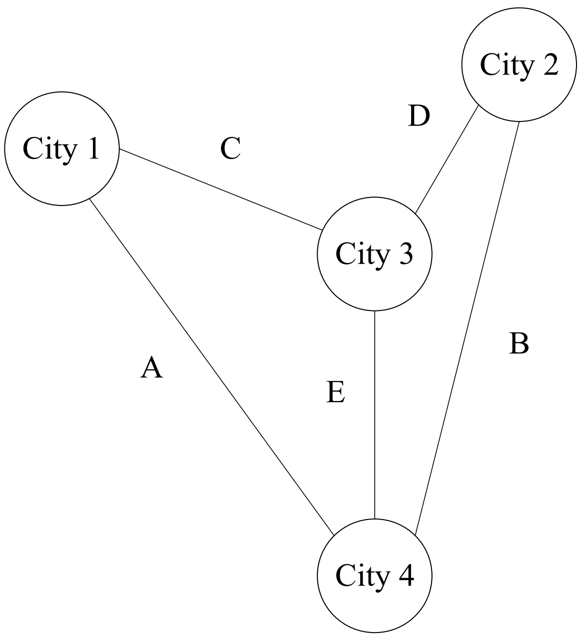

Figure 1. Example network for vehicle in platooning.

However, in case 2, suppose, for example, that vehicle 1 needs to travel from city 1 to city 4, and vehicle 2 needs to travel from city 2 to city 4, as per Figure 1. Assume that the shortest paths for vehicles 1 and 2 are A and B, respectively. If these two vehicles travel through the alternative routes C–E and D–E and are platooned in Section E, the distance that each vehicle travels increases, and the fuel-saving effect occurs in Section E. If the degree of fuel conservation in Section E is greater than that of the increased fuel consumption due to alternative routes, platooning with alternative routes can be preferred. In this case, not only the time schedule but also the fuel reduction rate affects the platooning decision.

In addition to the time schedule and fuel reduction, the density of the vehicle in the load network can affect the platooning. For example, if the slack time is enough in the time schedule and the number of vehicles crossing the same route is large, platooning through alternative routes may not be necessary, even if the fuel reduction rate is high. In this study, the effects of these variables, e.g., the time schedule, percentage of fuel reduction, and vehicle density, on vehicle platooning were examined. The following assumptions were made to simplify the problem considered in this study.

1. The schedule of each vehicle, including the earliest time of departure from the origin and the latest time of arrival at the final destination, is known and deterministic;
2. Each vehicle can wait at any node for the vehicle platooning if his/her time schedule is not violated;
3. The time distance for each link is constant; i.e., congestion due to the climate or accidents is not considered;
4. The fuel consumption of each vehicle is directly proportional to the time distance;
5. The fuel reduction rate is defined as the percentage of fuel reduction due to the platooning and is only applied for the following vehicles, i.e., the fuel reduction rate is set as 0 for the lead vehicle;
6. The fuel reduction rate resulting from platooning is constant for all the vehicles and links;
7. The vehicles are identical;
8. There is no time schedule violation.

The VRPP was formulated as a mixed-integer linear problem on the basis of the previous assumptions. The objective function of the VRPP was to minimize the total fuel cost.

3.2. Mathematical Model

Notation

$\|N\|$	number of nodes
$\|V\|$	number of vehicles
N	set of nodes $(h, i, j = 1, 2, 3, \ldots, \|N\|)$
V	set of vehicles $(k, l = 1, 2, 3, \ldots, \|V\|)$
L	set of arcs, $L = \{(i, j) : 1 \leq i, j \leq \|N\|, i \neq j\}$
$TFCI$	total fuel consumption of the initial solution
$TFCF$	total fuel consumption of the final solution
TNV	total number of vehicles
$TNVP$	total number of vehicles that are platooned with any other vehicles
$TNVD$	total number of vehicles that changed their route for a detour
NAP	total number of arcs platooned by vehicles
TNA	total number of arcs used by vehicles

Constants

s_k	origin node for the vehicle k, $s_k \in N$
d_k	destination node for the vehicle k, $d_k \in N$
DT_k	earliest departure time from s_k for vehicle k
AT_k	latest arrival time at d_k for vehicle k
$TD_{i,j}$	time distance between node i and node j
c	unit fuel cost per unit time distance
η	following vehicle's fuel reduction rate, resulting from vehicle platooning
δ	leading vehicle's fuel reduction rate
τ	slack time of vehicles
M	big number

Decision variables

$x^k_{i,j}$	binary variable; if vehicle k uses arc (i,j), 1; otherwise, 0
$y^{k,l}_{i,j}$	binary variable : if vehicles k and l are platooned on arc (i,j), 1; otherwise 0
$r^k_{i,j}$	number of vehicles platooned with vehicle k on arc (i,j)
$t^k_{i,j}$	arrival time at node j from node i of vehicle k when $x^k_{i,j}=1$. If $x^k_{i,j}=0$, $t^k_{i,j}=0$.
$TR_{i,j}$	total number of vehicles that achieve reduced fuel consumption by platooning on arc (i,j), i.e., total number of following vehicles in platooning groups on arc (i,j)
$z^k_{i,j}$	binary variable for linearization for $TR_{i,j}$
$p^k_{i,j}$	decision variable for linearization for $TR_{i,j}$

Variables for heuristics

L^k	set of routes, i.e., arcs (*i,j*), for vehicle k
$edt^k_{(i,j)}$	earliest possible departure time from origin of arc (*i,j*) for vehicle k
$ldt^k_{(i,j)}$	latest possible departure time from origin of arc (*i,j*) for vehicle k
$cdt^k_{(i,j)}$	actual departure time from origin of arc (*i,j*) for vehicle k
s^k	current solution for vehicle k, i.e., $\left\{edt^k_{(i,j)},\ ldt^k_{(i,j)},\ cdt^k_{(i,j)}\right\}$ for all i and j in L^k
$f(s^k)$	total fuel cos t for vehicle k when s^k is applied
u	random number generated from the uniform distribution U(0,1)

MIP Formulation

$$Minimize\ \sum_{\forall k\in V}\sum_{\forall(i,j)\in L} c\cdot TD_{i,j}\cdot x^k_{i,j} - \eta\cdot\sum_{\forall(i,j)\in L} c\cdot TD_{i,j}\cdot TR_{i,j} - \delta\cdot\sum_{\forall k\in V}\sum_{\forall(i,j)\in L} c\cdot TD_{i,j}\cdot z^k_{i,j} \tag{1}$$

$$\sum\nolimits_{\forall(h,i)\in L} x^k_{h,i} - \sum\nolimits_{\forall(i,j)\in L} x^k_{i,j} = 0 \text{ for } \forall k,\ \forall i \text{ when } i\neq s_k \text{ and } i\neq d_k \tag{2}$$

$$\sum\nolimits_{\forall(i,j)\in L} x^k_{i,j} = 1,\ \text{if } i = s_k \text{ for } \forall k \tag{3}$$

$$\sum\nolimits_{\forall(i,j)\in L} x^k_{i,j} = 1,\ \text{if } j = d_k \text{ for } \forall k \tag{4}$$

$$\sum\nolimits_{\forall(i,j)\in L} x^k_{i,j} = 0,\ \text{if } j = s_k \text{ for } \forall k \tag{5}$$

$$\sum\nolimits_{\forall(i,j)\in L} x^k_{i,j} = 0,\ \text{if } i = d_k \text{ for } \forall k \tag{6}$$

The model consists of nodes that are the intersection of roads. This structure contributes to simplifying the model via a reduction in the number of determinants.

The objective of the VRPP is to minimize the fuel cost, as described in (1). Especially, our model can consider a leading vehicle's fuel reduction independently. Equations (2)–(6) show the network balancing constraints. Equation (2) ensures that the inflow and outflow on each node of each vehicle are identical, except for the departure and destination nodes. Equations (3) and (4) indicate that vehicle k should leave its origin and arrive at its destination, respectively. Equations (5) and (6) indicate that vehicle k should not arrive at its origin and or leave its destination, respectively.

$$y^{k,l}_{i,j} - 1 \leq \frac{\left(t^k_{i,j} - t^l_{i,j}\right)}{M} \text{ for } \forall(i,j),\ \forall k,\ \forall l \text{ when } k\neq l, y^{k,l}_{i,j} \text{ is binary} \tag{7}$$

$$y^{k,l}_{i,j} - 1 \leq \frac{\left(t^l_{i,j} - t^k_{i,j}\right)}{M} \text{ for } \forall(i,j),\ \forall k,\ \forall l \text{ when } k\neq l, y^{k,l}_{i,j} \text{ is binary} \tag{8}$$

$$r^k_{i,j} = \sum\nolimits_{\forall l\in V, k\neq l} y^{k,l}_{i,j} \text{ for } \forall(i,j),\ \forall k \tag{9}$$

$$\sum\nolimits_{\forall l\in V, k\neq l} y^{k,l}_{i,j} \leq M\cdot x^k_{i,j} \text{ for } \forall(i,j),\ \forall k \tag{10}$$

Equations (7)–(10) show the vehicle-platooning constraints. Equations (7) and (8) ensure that if vehicles k and l are platooned on arc (i,j), the arrival times of both vehicles at node j are the same. Here, $r^{k1}_{i,j}$ in (9) represents the number of vehicles platooned with vehicle $k1$ on arc (i,j). Additionally, $r^{k1}_{i,j}$ represents the number of following vehicles in a platooned group because $r_{i,j}^{k1}$ does not include vehicle $k1$. Equation (10) indicates that vehicle $k1$ should travel arc (i,j) for platooning on arc (i,j).

If vehicle k does not have any platooned vehicle on arc (i,j), $r_{i,j}^{k}$ is zero. If vehicle k is platooned with at least one vehicle on arc (i,j), the total number of vehicles in the platooned group is $r_{i,j}^{k}+1$. Therefore, $r_{i,j}^{k}/\left(r_{i,j}^{k}+1\right)$ for all vehicles in the same platooned group is the number of following vehicles in multiple platooned groups on arc (i,j). Hence, $TR_{i,j}$ is calculated as follows:

$$TR_{i,j}=\sum\nolimits_{\forall k\in V}\left(r_{i,j}^{k}/\left(r_{i,j}^{k}+1\right)\right) \text{ for } \forall(i,j) \tag{11}$$

$$z_{i,j}^{k}\le\sum\nolimits_{\forall l\in V,k\neq l}y_{i,j}^{k,l} \text{ for } \forall(i,\ j)\ ,\ \forall k \tag{12}$$

$$z_{i,j}^{l}+z_{i,j}^{k}\le 1+M\ (1-y_{i,j}^{kl}) \text{ for } \forall(i,\ j)\ ,\ \forall k,\ l \text{ when } k\neq l \tag{13}$$

$$p_{i,j}^{k}\le r_{i,j}^{k} \text{ for } \forall(i,\ j)\ ,\ \forall k \tag{14}$$

$$p_{i,j}^{k}\le M\cdot z_{i,j}^{k} \text{ for } \forall(i,\ j)\ ,\ \forall k \tag{15}$$

$$TR_{i,j}=\sum\nolimits_{\forall k\in V}p_{ij}^{k}\text{for } \forall(i,\ j) \tag{16}$$

Equations (12)–(16) are considered for the linearization of (11). Equation (12) ensures that $z_{i,j}^{k}$ = 0 when vehicle k is not platooned with any vehicle on arc (i,j). Then, (13) ensures that $z_{i,j}^{k}$ = 0 except for one vehicle in a platooned group. Therefore, only one $p_{i,j}^{k}$ in a platooned group becomes $r_{i,j}^{k}$, and the other $p_{i,j}^{k}$ in the platooned group becomes 0 according to Equations (14) and (15) when $TR_{i,j}$ is maximized, i.e., to minimize (1).

$$t_{i,j}^{k}\le M\cdot x_{i,j}^{k} \text{ for } \forall(i,j),\ \forall k \tag{17}$$

$$t_{i,j}^{k}\ge\sum\nolimits_{\forall h}t_{h,i}^{k}+TD_{i,j}-M\cdot\left(1-x_{i,j}^{k}\right) \text{ for } \forall(i,j),\ \forall k \tag{18}$$

$$t_{i,j}^{k}\ge TD_{i,j}+DT_{k}-M\cdot\left(1-x_{i,j}^{k}\right) \text{ for } \forall(i,j),\ \forall k \text{ when } i=s_{k} \tag{19}$$

$$t_{i,j}^{k}\le AT_{k}+M\cdot\left(1-x_{i,j}^{k}\right) \text{ for } \forall(i,j),\ \forall k \text{ when } j=d_{k} \tag{20}$$

Equations (17)–(20) show the time schedule constraints. Here, $t_{i,j}^{k}$ is only positive when vehicle k travels arc (i,j) by the constraint in (17). Equation (18) indicates that the arrival time at node j should be equal to or longer than the sum of the arrival time at the previous node and the time distance of arc (i,j). Equation (19) ensures that vehicle k departs after DT_k at the origin. The constraint in (20) ensures that vehicle k arrives at the destination before AT_k.

$$x_{i,j}^{k}\in\{0,\ 1\} \text{ for } \forall(i,j),\ \forall k \tag{21}$$

$$y_{i,j}^{k,l}\in\{0,\ 1\} \text{ for } \forall(i,j),\ \forall k,\ l \tag{22}$$

$$z_{i,j}^{k}\in\{0,\ 1\} \text{ for } \forall(i,j),\ \forall k \tag{23}$$

$$t_{i,j}^{k}\ge 0 \text{ for } \forall(i,j),\ \forall k \tag{24}$$

Equations (21)–(24) show the range constraints for the decision variables.

Additionally, our model can avoid high congestion by adding more variables and constraints. Let (a,b) the time interval, e.g., (10,20), (20,30) and so on, and $w_{ij}^{k,(a,b)}$ the binary variable.

$$t_{i,j}^{k}>a-M\cdot\left(1-w_{i,j}^{k,(a,b)}\right)\ for\ \forall\ k \tag{25}$$

$$t_{i,j}^{k} \leq b + M \cdot \left(1 - w_{i,j}^{k,(a,b)}\right) \; for \; \forall \; k \tag{26}$$

$$\sum\nolimits_{\forall (a,b)} w_{ij}^{k,(a,b)} = x_{ij}^{k} \; for \; \forall \; k, \sum\nolimits_{\forall k \in V, \forall (a,b),} w_{ij}^{k,(a,b)} \leq n \tag{27}$$

However, in this study, we assume that the origin, destination, and allowable time of each vehicle are random, and therefore such high congestion hardly occurs.

The foregoing MIP problem is a well-known NP-hard problem. The next section presents the proposed greedy heuristic algorithm and the simulation results for the realistic network.

4. Proposed Heuristic for Optimizing VRPP

The number of decision variables of the VRPP significantly increases depending on the number of nodes, vehicles, and arcs. Many heuristics models are continuously designed for such complicated problems as a multi-layered greedy heuristic, ant colony system-improved gray wolf optimization, and simulated annealing genetic algorithm [28–32]. For practical computation, we propose a greedy heuristic to solve the VRPP. In this heuristic, each vehicle considers the possible earliest departure time and latest departure time, and actual departure time at each node of the given route for platooning. This set of information is called the vehicle time schedule (VTS).

The initial route for each vehicle is set as the shortest-distance (Time distance is used in this study) proposed by Dijkstra [33], and it is assumed that each vehicle departs from the origin at the earliest time possible. The initial actual departure time at each node on the route of each vehicle is set as the earliest departure time. The latest departure time is calculated in reverse from the latest arrival time at the destination.

The greedy heuristic consists of two platooning heuristics: Platooning Algorithm 1 and Platooning Algorithm 2. Platooning Algorithm 1 carries out platooning by changing only the time schedule of the vehicle without changing the route of the vehicle. Platooning Algorithm 2 carries out platooning using the alternative route of the vehicle. The pseudocode of the two algorithms is described in the boxes below.

4.1. Platooning Algorithm 1

In Platooning Algorithm 1, a vehicle k is randomly selected from V. The set of all the other vehicles is defined as V'. One of the links that the selected vehicle passes through is randomly selected as arc (*i,j*). Next, the set of vehicles that passes through the selected arc is defined as V'', and one of them (vehicle l $\in$ V'') is chosen randomly. If the platooning between vehicles k and l are impossible due to their VTSs, vehicle l is excluded form V'', and this process is iterated until V'' becomes a null set. If platooning is impossible until V'' becomes a null set, one of the arcs that have not been selected is chosen. Thus, if there are no vehicles left that can be platooned for all the arcs that vehicle k passes, the heuristic is terminated.

However, if platooning between vehicles k and l is possible, it is determined as to whether the schedule of the preceding vehicle matches that of the following vehicle or vice versa. If both are possible, the time schedule of two vehicles is determined at random with a 50% probability, and if the only option is possible, the feasible schedule is selected. When the platooning schedule is determined, the solution is updated, and the heuristic is terminated. The process is iterated until all vehicles have been selected.

Algorithm 1: Platooning

$V' \leftarrow V$
while $V' \neq$ **do**
  randomly select k in V';
  $V' \leftarrow V' - \{k\}$;
  while $L^k \neq \; ||$ s^k is not updated **do**
    randomly select arc (*i,j*) in L^k;
    $L^k \leftarrow L^k - \{\text{arc }(i,j)\}$;
    set V'' is subset of V' of which route has arc (*i,j*);
    while $V'' \neq$ **do**
      randomly select l in V'';
      $V'' \leftarrow V'' - \{l\}$;

12: **if** $cdt^k_{(i,j)} \in \left[edt^l_{(i,j)},\ ldt^l_{(i,j)}\right] \,||\, cdt^l_{(i,j)} \in \left[edt^k_{(i,j)},\ ldt^k_{(i,j)}\right]$ **then**
13: generate random number u;
14: **if** $u < 0.5$ **then**
15: $cdt^k_{(i,j)} \leftarrow cdt^l_{(i,j)}$;
16: update departure times at nodes from node i to destination of the vehicle k according to $cdt^k_{(i,j)}$;
17: **else**
18: $cdt^l_{(i,j)} \leftarrow cdt^k_{(i,j)}$;
19: change departure times at nodes from node i to destination of the vehicle l according to $cdt^l_{(i,j)}$;
20: **end if**
21: update s^k based on changed VTS;
22: **else if** $cdt^k_{(i,j)} \in \left[edt^l_{(i,j)},\ ldt^l_{(i,j)}\right]$ **then**
23: $cdt^k_{(i,j)} \leftarrow cdt^l_{(i,j)}$;
24: change departure times at nodes from node i to destination of the vehicle k based on $cdt^k_{(i,j)}$;
25: update s^k based on changed VTS;
26: **else if** $cdt^l_{(i,j)} \in \left[edt^k_{(i,j)},\ ldt^k_{(i,j)}\right]$ **then**
27: $cdt^l_{(i,j)} \leftarrow cdt^k_{(i,j)}$;
28: change departure times at nodes from node i to destination of the vehicle l based on $cdt^l_{(i,j)}$;
29: update s^k based on changed VTS;
30: **end if**
31: **end while**
32: **end while**
33: **end while**

4.2. Platooning Algorithm 2

In Platooning Algorithm 2, at first, vehicle k is randomly selected. The set of vehicles excluding the selected vehicle is called V''. Next, one of the arcs in the route of the selected vehicle is randomly selected. The set of arcs that vehicle k can move from the origin of the selected arc is called L', i.e., the set of detour arcs. Then, one of the arcs (*i,j*) in L' is randomly selected, and the shortest path among the routes from the origin of vehicle k to node j is found using the Dijkstra algorithm. This is added as a new route for vehicle k, and the VTS of vehicle k is updated accordingly. A vehicle that can be platooned is identified using the updated vehicle route and time schedule, as in Platooning Algorithm 1 (lines 5–32). If the solution provided by Platooning Algorithm 1 is better than the existing solution, the algorithm for the selected vehicle k is terminated, and the next vehicle in V' is selected. If the altered solution is not better than the existing solution, another detour arc of vehicle k in L' is selected, and the process is repeated until the solution is improved or all the detour arcs are selected.

Algorithm 2: Platooning

$V' \leftarrow V$
while $V' \neq$ **do**
$s^{k'} \leftarrow s^k$
randomly select vehicle k in V';
$V' \leftarrow V' - \{k\}$;
while $L^k \neq \varphi \,||\, s^k$ is not updated **do**
randomly select arc (*i,j*) in L^k;
$L^k \leftarrow L^k - \{\text{arc } (i,j)\}$;
set L' is the set of links for the feasible detour of arc (i,j) for vehicle k;
while $L' \neq$ **do**
randomly select arc (i,j) in L';
$L' \leftarrow L' - \{\text{arc } (i,j)\}$;
construct new route L^k (find shortest path to vehicle′s destination from the node j);
update VTS of L^k;
do Platooning Algorithm 1 (lines 5–32) for vehicle k;
if $f\left(s^{k'}\right) > f\left(s^k\right)$ **then**
go to out;
else

```
    s^k ← s^k';
    end if
  end while
 end while
out:
end while
```

The simulation procedure is described in the box below. T_1 is a counter to check the termination condition. T_2 is a counter to check the termination condition. T_1 and T_2 increase by 1 in each iteration, and T_1 is reset to 0 when the solution is improved in each iteration. The termination conditions are as follows. If the solution does not improve during 20 iterations or the maximum number of iterations (10,000) is reached, the simulation is terminated. After the solution is initialized in Step 2, Platooning Algorithms 1 and 2 are repeated in Steps 3 and 4, respectively. As described previously, in Step 2, the routes of all the vehicles are minimized using the Dijkstra algorithm, following which the platooning group is formed via Steps 3 and 4. Details regarding the simulation experiment and parameters are presented in Section 5. A simple example to show how our greedy heuristic works is in Appendix A. The simulation program was coded using Visual Basic (Visual Studio 2017) programming language. The simulation program was run on an Intel Core i7-7600 U central processing unit @ 2.80 GHz with 12 GB of random-access memory.

Simulation Procedure

Step 1: Let $T_1 = 0$; $T_2 = 0$
Step 2: Initialization

- Initial route is created by the Dijkstra algorithm for each vehicle.
- VTS is determined by the earliest departure time at the origin for each vehicle.
- Let the best solution be the current solution.

Step 3: Vehicle platooning with the same route

- Conduct Platooning Algorithm 1.

Step 4: Vehicle platooning with detour routes

- Conduct Platooning Algorithm 2.

Step 5: $T_1 = T_1 + 1$; $T_2 = T_2 + 1$
Step 6: If the best solution is updated at Step 3 or 4, let $T_1 = 0$.
Step 7: If $T_1 = 20$ or $T_2 = 10{,}000$, terminate the simulation procedure. Otherwise, go to Step 3.

5. Experiments and Results

5.1. Settings

The nodes and arcs of the networks for numerical examples were created according to actual data (corresponding to Germany, Japan, and Korea). First, several cities with a large population were selected, and the intersections among them were considered [34–36]. In the case of Germany, 21 nodes consisting of 10 cities (Berlin, Hamburg, Munich, Cologne, Frankfurt, Stuttgart, Dortmund, Hannover, Leipzig, and Nurnberg) and 11 intersections were selected. For Japan, 14 nodes consisting of eight cities (Tokyo, Yokohama, Osaka, Nagoya, Kobe, Kyoto, Fukuoka, and Kawasaki) and six intersections were selected. For Korea, 13 nodes consisting of seven metropolitan cities (Seoul, Busan, Incheon, Daegu, Daejeon, Gwangju, and Ulsan) and six intersections were selected. The cost of the arc was given on the basis of the time distance between the nodes. The time distances were determined via Google Maps for Germany and Japan and via the Naver map for South Korea [37,38].

Different scenarios with regard to the fuel reduction rates, number of vehicles, and time windows were considered, as shown in Table 4. In the examples, we assume that the fuel reduction rate of a leading vehicle is zero (δ = 0).

Table 4. Parameters of simulation experiment.

Parameter	Value
Fuel reduction rate (η)	0.05; 0.1; 0.15; 0.2; 0.25; 0.3
Number of vehicles ($\|V\|$)	25; 50; 100; 200
Slack time (τ)	3×60; 6×60; 12×60; 24×60 (min)

The earliest departure time from the origin for each vehicle was randomly generated from the uniform distribution with [0,24] (h). Hence, the latest arrival time at the destination for each vehicle was the sum of the earliest departure time at the origin for the vehicle, the shortest time distance between the origin and the destination, and the slack time shown in Table 4.

Each scenario was created according to the cases of three countries and the parameters in Table 4, and the simulation was performed 10 times with random generation of the origin and destination of each vehicle for the same scenario; i.e., 2880 ($3 \times 6 \times 4 \times 4 \times 10$) problems were designed and evaluated. Additionally, for each problem, the average fuel cost of three repetitions in the cases of Platooning Algorithms 1 and 2 was considered.

5.2. Results and Analyses

Table 5 shows the total cost and computation time of numerical examples. Detail parameter settings are as follows. The fuel reduction rate is 0.3; slack time is 60×3; the origin and destination of each vehicle are randomly selected; the departure time of each vehicle is randomly selected between 0 and 180.

Table 5. Total cost and computation time of numerical examples.

# of Nodes (Arcs)	# of Vehicles	Model	Problem					Total CPU Time (s)	Error
			1	2	3	4	5		
5 (12 arcs)	5	G	205	80.5	200.5	250	250	0.031	0.029
		O	205	80.5	184	250	238.5	16.786	
	10	G	385.5	334	539	533.5	467	0.031	0.056
		O	355.5	332.5	502.5	519	429	1382.155	
	15	G	729	774	917	744.5	548	0.094	
		O						(>12 h)	
10 (24 arcs)	5	G	397	172.5	295	178	282	0.068	0.017
		O	385	172.5	295	175	275	270.676	
	10	G	588.5	594	588.5	677	465	0.078	0.091
		O	520.5	521	573.5	612	444	24481.321	
	15	G	768.5	881	776	758	882.5	0.125	
		O						(>12 h)	
14 (57 arcs) (Republic of Korea)	5	G	879.2	855	966.7	744	1040	0.038	0.011
		O	859.1	855	940.9	744	1038.8	4842.662	
	10	G	1810.2	1615.7	1377.4	1576.7	2054.5	0.188	
		O						(>12 h)	
	15	G	2874.5	2997.6	3160.8	2915.3	2607.8	0.7160044	
		O						(>12 h)	

Table 5. *Cont.*

# of Nodes (Arcs)	# of Vehicles	Model	Problem					Total CPU Time (s)	Error
			1	2	3	4	5		
21 (92 arcs) (Germany)	5	G	675.5	1030	760.3	1145	1004	0.236	0.001
		O	675.5	1030	760.3	1140	1004	20729.978	
	10	G	1920.9	2143.2	1599	1981.7	2279	0.562	
		O						(>12 h)	
	15	G	2313.3	2689.1	3105.4	2561.6	2413.5	1.401	
		O						(>12 h)	

G: greedy heuristic, O: optimal.

The optimality gap, the ratio of the difference between the heuristic solution and optimal solution to the optimal solution, is, on average, 3.4%. It seems reasonable. In addition, the computation time of our greedy heuristic is only a few seconds in all cases. Our problem achieves less computation time compared to a similar previous study, e.g., the computation time of the heuristic by Larson et al. is almost 60–100 s [5]. Moreover, some previous studies using the Chicago network also limited the number of vehicles, e.g., 25 vehicles [14], or removed parts of nodes intentionally by pre-processing, e.g., Luo et al. (2018) [17].

Figure 2 shows the decrease in the total fuel consumption with the increasing heuristic iteration number for Germany, with the following parameters: fuel reduction rate = 0.1; number of vehicles = 200; slack = 24 × 60. With platooning, the total fuel consumption was reduced by approximately 15% compared with that of the initial solution (without platooning). The reason why the convergence is fast is that the opportunity for platooning is much due to the high density of vehicles and sufficient slack time. In such a similar condition, the convergence would be fast regardless of the number of nodes or arcs. On the contrary, when the density of vehicles is low, the convergence can be slow due to the lack of platooning opportunities. More results regarding with convergence of our greedy heuristic are shown in Appendix B.

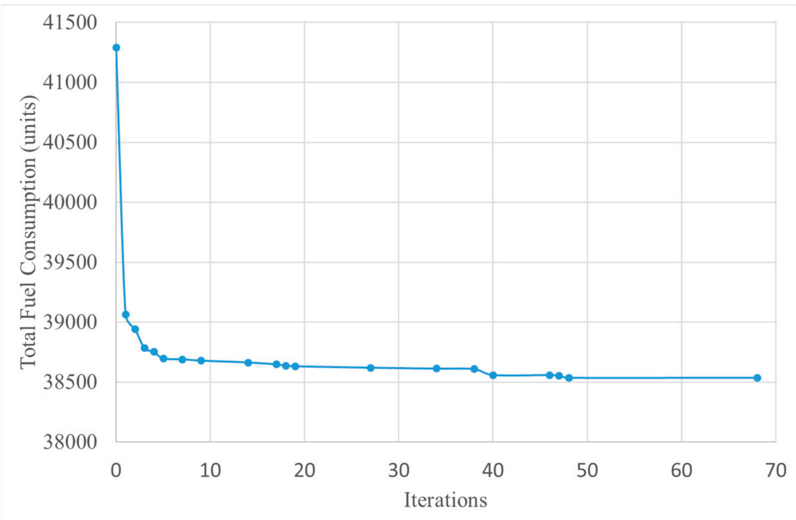

Figure 2. Total fuel consumption changes in each iteration for the example problem.

Tables 6–9 present the experimental results for each independent variable. The following four performance measures were used to evaluate the effects of the independent variables.

$$\% \text{ of fuel reduction} = \frac{(TFCI - TFCF)}{TFCI} \times 100\ (\%)$$
$$\% \text{ of platooned vehicles} = \frac{TNVP}{TNV} \times 100\ (\%)$$
$$\% \text{ of route} - \text{changed vehicles} = \frac{TNVD}{TNV} \times 100\ (\%)$$
$$\% \text{ of platooned arcs} = \frac{NAP}{TNA} \times 100\ (\%)$$

Table 6. Performance by country.

Country	% of Fuel Reduction	% of Platooned Vehicles	% of Route-Changed Vehicles	% of Platooned Arcs
Germany	6.94	64.36	6.69	60.68
Japan	9.95	77.38	7.09	61.64
Korea	7.98	62.19	4.49	62.98

Table 7. Performance for different fuel reduction rates.

Fuel Reduction Rate	% of Fuel Reduction	% of Platooned Vehicles	% of Route-Changed Vehicles	% of Platooned Arcs
0.05	2.10	68.97	2.31	59.61
0.1	4.36	68.61	3.99	60.52
0.15	6.77	68.21	5.52	61.44
0.2	9.34	67.82	6.75	62.12
0.25	12.10	67.47	8.32	63.08
0.3	15.06	66.81	9.66	63.84

Table 8. Performance for different numbers of vehicles.

Number of Vehicles	% of Fuel Reduction	% of Platooned Vehicles	% of Route-Changed Vehicles	% of Platooned Arcs
25	4.83	48.78	5.82	43.00
50	7.22	63.28	6.91	57.72
100	9.71	75.48	6.27	70.32
200	11.39	84.39	5.37	76.03

Table 9. Performance for different slack times.

Slack Time (min)	% of Fuel Reduction	% of Platooned Vehicles	% of Route-Changed Vehicles	% of Platooned Arcs
180	5.20	54.02	5.63	45.53
360	7.02	64.95	6.20	56.82
720	8.97	73.91	6.43	66.90
1440	11.96	79.04	6.10	77.83

The percentage of route-changed vehicles is the percentage of vehicles that changed routes from the shortest path for the purpose of platooning. The percentage of platooned arcs can indicate the extent to which the road capacity can be increased.

Table 6 shows the difference in performance measures among the countries. There are differences in platooning results according to network characteristics, such as the number of nodes, average distance between nodes, and ratio between the number of nodes and number of arcs. The reason for the performance differences was not investigated in the present study and is intended to be researched in the future.

Table 7 presents the simulation results according to the fuel reduction rates. As the fuel reduction rate increased, the percentage of fuel reduction increased, which is intuitive. Additionally, as the fuel reduction rate increased, the percentage of vehicles that were detoured for platooning increased from 2.31% to 9.66%. Consequently, the percentage of platooned arcs increased by approximately 4%. Thus, if the fuel reduction rate increases, finding alternative routes for vehicle platooning becomes more important.

Table 8 presents the experimental results according to the number of vehicles in the network. Increasing the number of vehicles increased the vehicle density in the network under the same conditions. When the vehicle density increased, more vehicles had the opportunity to group together. Therefore, the total fuel reduction effect increased, and the proportion of platooned vehicles increased rapidly (an increase from 4.83% to 11.38% in fuel reduction and an increase from 48.78% to 84.39% in the percentage of platooned vehicles). Additionally, the percentage of platooned arcs increased by 33%.

However, the percentage of route-change vehicles changed. When the vehicle density is too low, platooning becomes rare; thus, the percentage of route-changed vehicles becomes low. When the vehicle density is too high, vehicles do not need to select an alternative route. Thus, the percentage of route-changed vehicles becomes low.

Table 9 presents the simulation results for different slack times. As the slack time increased, the percentage of fuel reduction, percentage of platooned vehicles, and percentage of platooned arcs increased from 5.2% to 11.96%, from 54.02% to 79.04%, and from 45.53% to 77.83%, respectively. This is because the opportunity for platooning increased as the time schedule of the vehicles became more relaxed.

The effect of the slack time on the percentage of route-changed vehicles was identical to that in the case of different numbers of vehicles (Table 8, described above). In addition, as shown in Tables 8 and 9, increases in the number of vehicles and the slack time significantly increased the proportion of platooning vehicles.

Figures 3 and 4 show the percentage of fuel reduction and percentage of platooned arcs for different numbers of vehicles and slack times. As shown in Figure 3, the percentage of fuel reduction increased by >15% when there were 200 vehicles and the slack time was 1440 min, when compared with the case of 25 vehicles and a slack time of 180 min. This is an average for all fuel reduction rate levels and may increase further depending on fuel reduction rate level. Thus, considerable fuel reduction can be achieved if optimal platooning is employed. As shown in Figure 4, the percentage of platooned arcs increased from 25% to ≥80% when there were 200 vehicles and a slack time of 1440 min, when compared with the case of 25 vehicles and a slack time of 180 min. As shown in Figure 4, 80% of the arcs that vehicles passed through were platooned, indicating that platooning can create space on the road, improving the traffic flow.

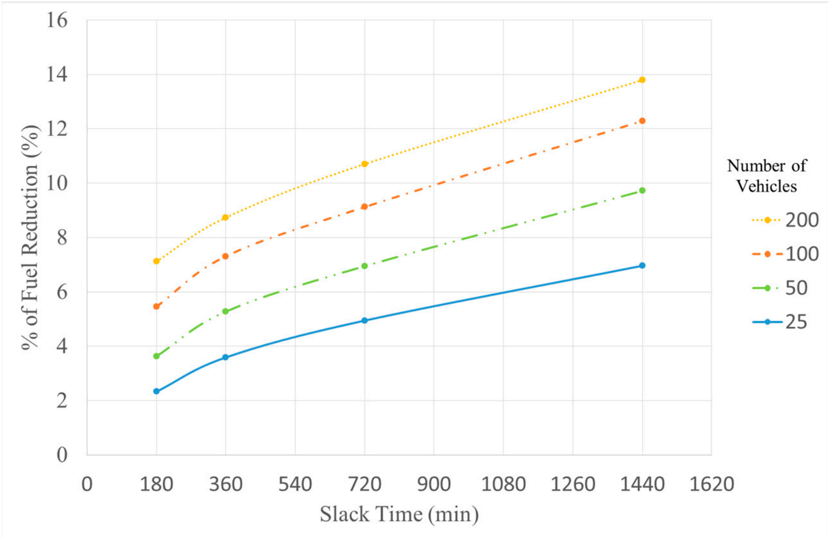

Figure 3. Percentage of fuel reduction for different numbers of vehicles and slack times.

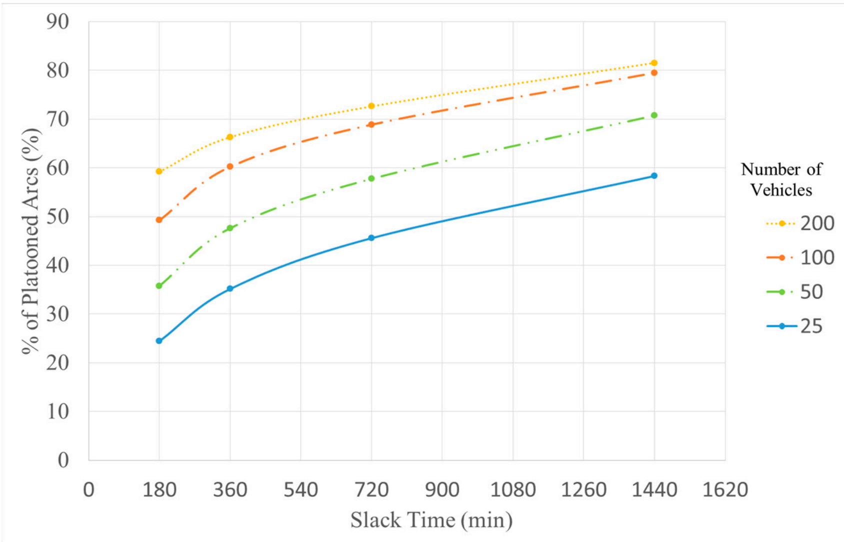

Figure 4. Percentage of platooned arcs for different numbers of vehicles and slack times.

Based on the results in Figures 3 and 4, we can choose the best option between the increasing number of vehicles and relaxing vehicle schedules depending on the actual situation.

6. Conclusions and Future Research

VRPP reduces the air resistance to subsequent vehicles, and therefore, the fuel consumption and the air pollution of the vehicles can be reduced. Additionally, platooning decreases the risk of accidents and helps traffic congestion. In this study, we propose a practical VRPP mathematical model with a deadline for vehicles, continuous-time units, traffic congestion avoidance, and heterogeneous vehicles. For reasonable computation time, a greedy heuristic is suggested. A greedy heuristic consists of Platooning Algorithm 1 and Platooning Algorithm 2. The performance of the heuristic using four measures: the percentage of fuel reduction, percentage of platooned vehicles, percentage of route-changed vehicles, and percentage of platooned arcs, with different fuel reduction rates, numbers of vehicles, and slack times. In the numerical examples, the following results were obtained.

(1) The optimality gap by our model, the ratio of the difference between the heuristic solution and optimal solution to the optimal solution, is, on average, 3.4%. The model by Larson et al. obtained the 1% optimality gap in small problems within 60–100 s;
(2) As the fuel reduction rate increased, a larger proportion of vehicles were platooned via alternative routes. As the fuel reduction rate increased, the percentage of vehicles detoured for platooning increased from 2.31% to 9.66%. The percentage of platooned arcs increased by approximately 4%;
(3) As the number of vehicles and slack time increased, the proportion of platooned vehicles increased. The percentage of fuel reduction increased more than 15% when there were 200 vehicles, and the slack time was 1440 min, compared with the case of 25 vehicles and a slack time of 180 min;
(4) Increasing the number of platooned vehicles saved space on the road;
(5) In the future study, a more in-depth analysis of the relationship between the structure of nodes and arcs and the detouring tendency of platooned vehicles is planned.

Author Contributions: Conceptualization, S.P. and C.H.; methodology, S.P. and H.S.; software, S.P.; validation, C.H. and H.S.; writing—original draft preparation, S.P. and C.H.; writing—review and editing, H.S.; funding acquisition, H.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (no. 2022R1F1A1065607 and no. 2020R1F1A1066272).

Data Availability Statement: The authors confirm that the data supporting the findings of this study are available within the article.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Tables A1–A3 show the example in the case of 5 nodes and 5 vehicles.

Table A1. The fuel cost of nodes (5 nodes and 5 vehicles example).

Node	1	2	3	4	5
1		5			100
2	5		5	50	
3		5			100
4		50			50
5	100		100	50	

Table A2. The origin, destination, and deadline of vehicles (5 nodes and 5 vehicles example).

Vehicle	**Origin**	**Destination**	**Deadline**
1	3	1	177.2948
2	5	3	19.54208
3	1	2	4.058831
4	4	5	29.75503
5	4	2	8.943096

Table A3. The initial and final solutions of greedy heuristics.

Vehicle	**Initial Solution**	**Final Solution**
1	3→1	3→1
2	5→3	5→4→2→3

Table A3. *Cont.*

Vehicle	Initial Solution	Final Solution
3	1→2	1→2
4	4→5	4→5
5	4→2	4→2
Total cost	215	205

The total cost of the optimal solution for this example is 205 (by Gurobi). In our greedy heuristic, the initial schedule of each vehicle is determined by its shortest path, as shown in Table A3. The total cost of this case is 215. Next, each vehicle considers the detour randomly if the detour is satisfied with existing constraints. Then, if the fuel reduction is possible by the detour, change the path accordingly. Finally, the second vehicle changes the path as follows: 5-4-2-3 with a total cost of 205. In such a way, our greedy heuristic can quickly reduce the total cost.

Appendix B

Figure 2 and Table 5 in Section 5 show how quickly our greedy heuristic finds the solution. Table A4 also shows that more than half of cases quickly converge within 40 iterations, and all cases converge within 200 iterations.

Table A4. The iteration number of three countries.

Country	Min. Iteration	Avg. Iteration	Max. Iteration	Standard Deviation of Iteration
Germany	20	37.006	131	17.470
Japan	20	31.069	110	13.617
Korea	20	33.131	155	15.079

Table A5 shows more detailed results for each country. The final solutions of all countries are significantly improved even within small iterations, as in Table A4.

Table A5. Comparison of three countries' solutions.

Country	Avg. Initial Solution	Std. Initial Solution	Avg. Final Solution	Std. Final Solution
Germany	18,632.9	135,22.29	16,978.99	12,034.14
Japan	26,699.2	19,327.19	23,752.9	17,021.86
Korea	17,260.75	12,384.05	15,525.39	10,829.63

References

1. Scania Group. Platooning—Automated Driving. Available online: https://www.scania.com/group/en/platooning-automated-driving-for-fuel-savings/ (accessed on 2 January 2021).
2. Tech-F.A.Q. Vehicle Platooning. Available online: http://www.tech-faq.com/vehicle-platooning.html (accessed on 2 January 2021).
3. Larsson, E.; Sennton, G.; Larson, J. The vehicle platooning problem: Computational complexity and heuristics. *Transp. Res. C* **2015**, *60*, 258–277. [CrossRef]
4. Nourmohammadzadeh, A.; Hartmann, S. The fuel-efficient platooning of heavy duty vehicles by mathematical programming and genetic algorithm. *Lect. Notes Comput. Sci.* **2016**, *10071*, 46–57. [CrossRef]
5. Larson, J.; Munson, T.; Sokolov, V. Coordinated platoon routing in a metropolitan network. In Proceedings of the Seventh SIAM Workshop on Combinatorial Scientific Computing, Albuquerque, NM, USA, 10–12 October 2016; pp. 73–82. [CrossRef]
6. Zhao, L.; Sun, J. Simulation framework for vehicle platooning and car-following behaviors under connected-vehicle environment. *Procedia Soc. Behav. Sci.* **2013**, *96*, 914–924. [CrossRef]

7. Segata, M.; Dressler, F.; Lo Cigno, R.; Gerla, M. A simulation tool for automated platooning in mixed highway scenarios. In Proceedings of the 18th Annual International Conference on Mobile Computing and Networking; ACM Press: New York, NY, USA, 2012; Volume 16, p. 389. [CrossRef]
8. Nasri, M.I.; Bektaş, T.; Laporte, G. Route and speed optimization for autonomous trucks. *Comput. Oper. Res.* **2018**, *100*, 89–101. [CrossRef]
9. Dukkanci, O.; Kara, B.Y.; Bektaş, T. The green location-routing problem. *Comput. Oper. Res.* **2019**, *105*, 187–202. [CrossRef]
10. Demir, E.; Bektaş, T.; Laporte, G. An adaptive large neighborhood search heuristic for the pollution-routing problem. *Eur. J. Oper. Res.* **2012**, *223*, 346–359. [CrossRef]
11. Kramer, R.; Maculan, N.; Subramanian, A.; Vidal, T. A speed and departure time optimization algorithm for the pollution-routing problem. *Eur. J. Oper. Res.* **2015**, *247*, 782–787. [CrossRef]
12. Bhoopalam, A.K.; Agatz, N.; Zuidwijk, R. Planning of truck platoons: A literature review and directions for future research. *Transport. Res. Part B Methodolog.* **2018**, *107*, 212–228. [CrossRef]
13. Larson, J.; Kammer, C.; Liang, K.Y.; Johansson, K.H. Coordinated route optimization for heavy duty vehicle platoons. In Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), The Hague, The Netherlands, 6–9 October 2013; pp. 1196–1202.
14. Sokolov, V.; Larson, J.; Munson, T.; Auld, J.; Karbowski, D. Platoon formation maximization through centralized routing and departure time coordination. *arXiv* **2017**, arXiv:1701.01391. [CrossRef]
15. Nourmohammadzadeh, A.; Hartmann, S. Fuel-efficient truck platooning by a novel meta-heuristic inspired from ant colony optimization. *Soft Comput.* **2019**, *23*, 1439–1452. [CrossRef]
16. Larson, J.; Liang, K.-Y.; Johansson, K.H. A distributed framework for coordinated heavy-duty vehicle platooning. *IEEE Trans. Intell. Transport. Syst.* **2015**, *16*, 419–429. [CrossRef]
17. Luo, F.; Larson, J.; Munson, T. Coordinated platooning with multiple speeds. *Transp. Res. Part C Emerg. Technol.* **2018**, *90*, 213–225. [CrossRef]
18. Laporte, G. Fifty years of vehicle routing. *Transp. Sci.* **2009**, *43*, 408–416. [CrossRef]
19. Leung, S.C.; Zhang, Z.; Zhang, D.; Hua, X.; Lim, M.K. A meta-heuristic algorithm for heterogeneous fleet vehicle routing problems with two-dimensional loading constraints. *Eur. J. Oper. Res.* **2013**, *225*, 199–210. [CrossRef]
20. Bräysy, O.; Gendreau, M. Vehicle routing problem with time windows, Part II: Metaheuristics. *Transp. Sci.* **2005**, *39*, 119–139. [CrossRef]
21. Gendreau, M.; Tarantilis, C.D. *Solving Large-Scale Vehicle Routing Problems with Time Windows: The State-of-the-Art*; Cirrelt: Montreal, QC, Canada, 2010.
22. Tasan, A.S.; Gen, M. A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and deliveries. *Comput. Ind. Eng.* **2012**, *62*, 755–761. [CrossRef]
23. Pradenas, L.; Oportus, B.; Parada, V. Mitigation of greenhouse gas emissions in vehicle routing problems with backhauling. *Expert Syst. Appl.* **2013**, *40*, 2985–2991. [CrossRef]
24. Brar, G.S.; Saini, G. Milk run logistics: Literature Review and directions. In Proceedings of the World Congress on Engineering, London, UK, 6–8 July 2011; Volume 1, pp. 797–801.
25. Montoya-Torres, J.R.; Franco, J.L.; Isaza, S.N.; Jiménez, H.F.; Herazo-Padilla, N. A literature review on the vehicle routing problem with multiple depots. *Comput. Ind. Eng.* **2015**, *79*, 115–129. [CrossRef]
26. Campbell, A.M.; Wilson, J.H. Forty years of periodic vehicle routing. *Networks* **2014**, *63*, 2–15. [CrossRef]
27. Gulczynski, D.; Golden, B.; Wasil, E. The period vehicle routing problem: New heuristics and real-world variants. *Transp. Res. E Logist. Transp. Rev.* **2011**, *47*, 648–668. [CrossRef]
28. Karimi-Mamaghan, M.; Mohammadi, M.; Pasdeloup, B.; Meyer, P. Learning to select operators in meta-heuristics: An integration of Q-learning into the iterated greedy algorithm for the permutation flowshop scheduling problem. *Eur. J. Oper. Res.* **2023**, *304*, 1296–1330. [CrossRef]
29. Wang, M.; Jiang, H.; Li, Y.; Wu, S.; Xia, L. Location determination of hierarchical service facilities using a multi-layered greedy heuristic approach. *Eng. Optim.* **2022**, 1–15. [CrossRef]
30. Lu, F.; Feng, W.; Gao, M.; Bi, H.; Wang, S. The fourth-party logistics routing problem using ant colony system-improved grey wolf optimization. *J. Adv. Transp.* **2020**, *2020*, 8831746. [CrossRef]
31. Lu, F.; Bi, H.; Huang, M.; Duan, S. Simulated annealing genetic algorithm based schedule risk management of IT outsourcing project. *Math. Probl. Eng.* **2017**, *2017*, 6916575. [CrossRef]
32. Wen, H.; Wang, S.X.; Lu, F.Q.; Feng, M.; Wang, L.Z.; Xiong, J.K.; Si, M.C. Colony search optimization algorithm using global optimization. *J. Supercomput.* **2022**, *78*, 6567–6611. [CrossRef]
33. Dijkstra, E.W. A Note on Two Problems in Connexion with Graphs. *Numer. Math.* **1959**, *1*, 269–271. [CrossRef]
34. Available online: https://worldpopulationreview.com/countries/cities/germany (accessed on 2 January 2023).
35. Available online: https://worldpopulationreview.com/countries/cities/japan (accessed on 2 January 2023).
36. Available online: https://worldpopulationreview.com/countries/cities/south-korea (accessed on 2 January 2023).
37. Available online: https://www.google.com/maps (accessed on 2 January 2023).
38. Available online: https://map.naver.com/ (accessed on 2 January 2023).