applied sciences

MDPI

Article

A Location Routing Problem with Time Windows Consideration: A Metaheuristics Approach

Meilinda Fitriani Nur Maghfiroh [1,2], Vincent F. Yu [2,3], Anak Agung Ngurah Perwira Redi [4] and Bayu Nur Abdallah [5,*]

[1] Department of Industrial Engineering, Universitas Islam Indonesia, Yogyakarta 55584, Indonesia
[2] Department of Industrial Management, National Taiwan University of Science and Technology, Taipei 106, Taiwan
[3] Center for Cyber-Physical System Innovation, National Taiwan University of Science and Technology, Taipei 106, Taiwan
[4] Department of Industrial Engineering, Sampoerna University, Jakarta 12780, Indonesia
[5] Department of Digital Business, Institut Teknologi Kalimantan, Balikpapan 76127, Indonesia
* Correspondence: bayunur@lecturer.itk.ac.id

Abstract: A substantial distribution system is achievable to attain through the simultaneous design of the location facility and route generation. This problem is generally called a location routing problem (LRP), where both strategic decision and operational decisions are considered during the decision initial phase. Nevertheless, for some real applications, a consideration of time restriction due to several reasons should be included in the model. Some examples include the distribution system of perishable goods, a company's objective towards a high level of customer service, or emergency/disaster situations. Consequently, this research incorporates the location routing problem (LRP) with time window constraints, including the working hour and also time restriction in each customer. The location routing problem with time windows (LRPTW) could not be solved easily, especially for a large-sized problem and or one which is considered an NP-hard problem. In consideration with its large solution pools, this paper suggests an integration of variable neighborhood search (VNS) with path relinking (PR) to solve LRPTW. The method is first applied in the LRP benchmark to validate the effectiveness and the robustness of the proposed method. Computational results clarify that VNSPR can achieve some of the best and most competitive solutions for the LRP when compared to existing approaches. Further, we developed LRPTW instances from the best solution found in LRP datasets. The proposed VNSPR method is also able to yield nearly optimal solutions for LRPTW instances.

Keywords: location routing problem; time windows; variable neighborhood search; path relinking

check for updates

Citation: Maghfiroh, M.F.N.; Yu, V.F.; Redi, A.A.N.P.; Abdallah, B.N. A Location Routing Problem with Time Windows Consideration: A Metaheuristics Approach. *Appl. Sci.* **2023**, *13*, 843. https://doi.org/10.3390/app13020843

Academic Editor: Vincent A. Cicirello

Received: 4 December 2022
Revised: 30 December 2022
Accepted: 31 December 2022
Published: 7 January 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The configuration of distribution systems is vital for most organizations because it generally necessitates a significant capital expenditure. A large portion of an organization's expenses is spent on distribution systems, indicating the need for low logistics costs [1]. Among the many decisions that need to be made for configuring distribution systems are location determination and route generation. Facility location determination and route generation have previously been resolved and conducted at differing levels [2], whereas routing can be undertaken more regularly during the operational stage and the facility must be established prior to this stage as it is part of long-term strategic planning. Nevertheless, many studies have developed an integrated model due to its high dependency among those two problems.

The location routing problem (LRP) consists of strategic and tactical decisions, in which the location to open and the distribution route must be determined simultaneously. LRP has been employed for making such decisions for many years in response to the needs of real-world applications. Substantial productivity benefits can be realized as a result of

the development of location routing models, which can determine true least-cost solutions to logistical problems while accounting for both strategic policy (facility location) and operational decisions (vehicle routing) [3].

Many variants of LRP have been widely studied depending on type of industry and the problems it is dealing with. The LRP's extensions include pickup and delivery LRP [4–6], open LRP [7–9], multi-objectives LRP [10–14], multi-echelon LRP [11,15–18], inventory LRP [19–24], and LRP with uncertainty [23,25–28]. Nevertheless, considering that in practice the customers must be served before a certain time, many studies have added an additional constraint: time windows. The applications for LRP with time windows are quite wide, including for commercial logistics application [27,29,30], humanitarian logistics application [31,32], green and/or sustainable logistics [33–35], and waste management [36,37]. Gündüz [38] introduced a single stage LRP with time windows (SSLRPTW) which combined capacitated facility location problem (CFLP) and VRPTW. The author tried to minimize the cost of routing and depot opening by decomposing the problem into two sub problems: CFLPTW and VRPTW. In the first phase, he solved the CFLPTW by assigning the customers to a depot and in the second phase, he used a tabu search to improve the solution. He extended the Solomon VRPTW benchmark into SSLRPTW benchmark. Zarandi et al. [28] solved LRPTW with fuzzy travel time. They developed an algorithm called the simulation algorithm by incorporating a simulation module in SA. Small instances problems from Prins et al. [39,40] and Barreto [41] were solved. Schiffer and Walther [33] introduced the electric location routing problem with time windows and partial recharging (ELRP-TWPR) to model the distribution network using electric vehicle. Similar to Gunduz [38], they also derived the problem based on VRPTW instances and added some constraints to model the recharging stage of electric vehicles.

The distribution system of chain stores, cold-chain, and relief goods distribution can be represented as LRP with time windows problem, where each vehicle has a service level agreement within the distribution center and customers (retailers, shelters, stores, etc.). Each customer has to be served before a certain desired time. Each driver also has a certain working time which is considered as a constraint as well. Other practices include the flood evacuation network and distribution with electric vehicles. This research extends the LRP model into the location routing problem with time windows (LRPTW) with goals to determine: (i) the depots to open, (ii) the customers, and (iii) the vehicles to be assigned to each open depot and the routes to be created to satisfy the demand within restricted time windows of the customers with the smallest cost.

This paper tackles both location and routing problems together with consideration of time windows and capacity restrictions at depots and vehicles for different size instances. This study is also the first attempt to create an instance of an LRPTW problem from LRP instances. The heuristic variable neighborhood search with path relinking (VNSPR) is presented as a method to solve the LRPTW. One key contribution of this paper is initiating a basis for evaluating the performance of VNSPR which demonstrates capability of solving CLRP and its extensions. This paper is organized as follows: Section 1 provide introduction and reviews previous studies on the CLRP, VRPTW and the related problems, Section 2 consist of two part, first presents the problem definition and mathematical formulation of the LRPTW, second gives an overview of the VNS and PR algorithm and introduces the proposed VNSPR heuristic for solving the LRPTW, while Section 3 presents the benchmark instances and algorithm verification, Section 4 provide discussion on the propose VNSPR to solve LRPTW, Section 5 draws conclusions and points out possible future research directions.

2. Materials and Methods

2.1. Problem Description

We describe the location routing problem with time windows in this subsection. There are m depot, k vehicles, and n customers. Each prospective depot has different capacity and opening cost. Each vehicle has the same capacity, and it leaves the open depot to deliver

goods to customers and go back to its original depot. Every customer must be served exactly once by one vehicle within its available service time window, which is bounded by an earliest time and a latest time. If a vehicle reaches a customer before the customer's service time window starts, then it must wait until the earliest service time. On the other hand, if a vehicle reaches a customer later than the customer's latest available service time, then the vehicle can no longer serve the customer. The customer has to be served by another vehicle. After arriving at a customer's location, the vehicle has to stay for a period of time, i.e., service time. The problem determines the location of the depots and vehicle routes to minimize total cost associated with establishing the depots and distributing the goods.

Given a graph $G = (V, L)$ in which V and L comprise the graph's vertices and links, respectively, V entails a subset I of m prospective depot sites and a subset $J = V \backslash I$ of n, the node of customers. Each set of nodes in V is connected by link E. For each link $(i, j) \in L$, a traveling cost c_{ij} is incurred to express the cost between node i and j. Every depot $i \in I$ has a fixed capacity W_i with its associated cost of opening O_i. Each node $j \in J$ holds a demand d_j that needs to be satisfied by vehicle k. The maximum number of K vehicles is available with its respected capacity Q. Each vehicle used from depot i will have a fixed cost F_i.

All nodes, customers, and depots are associated with time windows. The time window of a node, i, is specified by an interval $[e_i, l_i]$ where e_i and l_i represent the earliest and the latest arrival time, respectively. All vehicles must come to customer node i before the latest time windows l_i. Each depot will have time windows $[e_0, l_0]$, which indicate the working hour for the driver, from leaving the depot until the vehicle returns to the depot. The service/routing can only be performed within this range. The travel time between links $(i, j) \in L$ is denoted by t_{ij}. a_{ik} is the arrival time of vehicle k at customer i and p_{ik} is the departure time at customer i. Each vehicle k has an identical duration B_k equal to the length of a driver's working hours. The service time in each node i, δ_i, is a fixed number indicating the time needed to serve customer i. A vehicle can only serve customer i if the vehicle arrives within its time window, otherwise the waiting time w_i will be incurred. If the vehicle arrives at customer i within its time windows, $w_i = 0$.

Variables:

$$y_i = \begin{cases} 1, & \text{if the depot } i \text{ is opened;} \\ 0, & \text{otherwise.} \end{cases}$$

$$f_{ij} = \begin{cases} 1, & \text{if customer } j \text{ is assigned to an open depot } i; \\ 0, & \text{otherwise.} \end{cases}$$

$$x_{ijk} = \begin{cases} 1, & \text{if customer } j \text{ is visited immediately after customer } i \text{ by vehicle } k; \\ 0, & \text{otherwise.} \end{cases}$$

Objective function:

$$\min z = \sum_{i \in I} O_i y_i + \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} c_{ij} x_{ijk} + \sum_{k \in K} \sum_{i \in I} \sum_{j \in J} F_i x_{ijk} \tag{1}$$

subject to

$$\sum_{k \in K} \sum_{i \in V} x_{ijk} = 1 \qquad \forall j \in J \tag{2}$$

$$\sum_{j \in J} \sum_{i \in V} x_{ijk} d_j \leq Q \qquad \forall k \in K \tag{3}$$

$$\sum_{j \in J} d_i f_{ij} \leq W_i y_i \qquad \forall i \in I \tag{4}$$

$$\sum_{j \in V} x_{ijk} - \sum_{j \in V} x_{jik} = 0 \qquad \forall i \in V, k \in K \tag{5}$$

$$\sum_{i \in I} \sum_{j \in J} x_{ijk} \leq 1 \qquad \forall k \in K \tag{6}$$

$$\sum_{i\in S}\sum_{j\in S} x_{ijk} \le |S| - 1 \qquad \forall S \subseteq J, k \in K \tag{7}$$

$$\sum_{u\in J} x_{iuk} - \sum_{u\in V\{j\}} x_{ujk} \le 1 + f_{ij} \qquad \forall i \in I, j \in J, k \in K \tag{8}$$

$$\sum_{i\in V}\sum_{j\in V} t_{ij}x_{ijk} + \sum_{i\in V}(\delta_i + w_i)\sum_{j\in V} x_{ijk} \le B_k \qquad \forall k \in K \tag{9}$$

$$\sum_{k\in K}\sum_{i\in V} x_{ijk}\left(a_{ik} + t_{ij} + \delta_i + w_i\right) \le a_{jk} \qquad \forall j \in J \tag{10}$$

$$e_i \le a_{ik} \le l_i \qquad i \in V, k \in K \tag{11}$$

$$a_0 = w_0 = \delta_0 \tag{12}$$

$$y_i \in \{0,1\}, \forall i \in I \tag{13}$$

$$f_{ij} \in \{0,1\}, \forall i \in I, j \in V \tag{14}$$

$$x_{ijk} \in \{0,1\}, \forall i \in I, j \in J, k \in K \tag{15}$$

Equation (1) stipulates the objective function of this problem which is to minimize total cost for depot opening cost and distribution cost. Constraint (2) ensures that each customer belongs to exactly one route. Constraints (3) and (4) are capacity constraints for the route and depot while Constraints (5) and (6) guarantee that the route starts and ends in the same depot. Constraint (7) is a sub-tour elimination constraint. The sub-tour elimination constraint is used to ensure the connectedness of solutions. The sub-tour elimination constraint being used is based on the one presented in Yu et al. [3]. The subset S represents a sub-tour that constructs the solution for the problem. Thus, Constraint (7) imposes the connectivity of the sub-tour at each vehicle based on the cardinality of each subtour. Constraint (8) guarantees a customer is assigned to a depot if there is a route connecting them. The next constraints explain time window constraints. Constraint (9) is the maximum travel time constraint and Constraints (10)–(12) ensure the relations between arrival time, departure time, and service time with regard to a customer's time window. Finally, Constraints (13)–(15) provide details on the binary variables that were incorporated into the formulation.

2.2. Proposed Method

Hansen and Mladenović [42] proposed an algorithm that can minimize the probability of tumbling into local optima traps by systematically alter the surrounding area during the local search. This results in the creation of a metaheuristic that is known as variable neighborhood search (VNS), which has been further developed through its many extensions. Variable neighborhood search (VNS) has been recognized by its ability to solve various types of problems, especially the vehicle routing problem and its extensions [43–48]. VNS also has been proven to solve several cases of LRP including CLRP [49], inventory LRP [50], multi-echelon LRP [51], and periodic LRP [52]. The amended and intense VNS, i.e., variable neighborhood descent (VND), was employed by [53–55] to take advantage of different neighborhoods structures for the vehicle routing problem.

This section presents a variable neighborhood search with a path relinking (PR) algorithm developed for solving CLRP as used by Prins et al. [39]. The proposed VNSPR algorithm is explained in detail in the subsequent sections, including the general structure, solution representation, initial solution, different kinds of neighborhoods, and the detailed method.

2.2.1. General Structure of VNSPR

VNSPR is a hybridization of a VNS with a PR procedure. It holds benefits from both algorithms. Figure 1 depicts the elements and overall structure of the presented VNSPR. The VNS adjusted and transformed the neighborhood during the search procedure to

mitigate the solution being entrapped in the local optimal solution found. Initiating with an existing solution, the VNS seeks within a subset of neighborhoods, each of which is investigated using a local search procedure. The hybrid metaheuristic includes a variable neighborhood descent (VND) section to enrich the operation of the VNS. PR is used in various strategies to incorporate search expansion and solution diversity. This method creates new solutions by tracing paths that unite superior quality solutions to elite solutions.

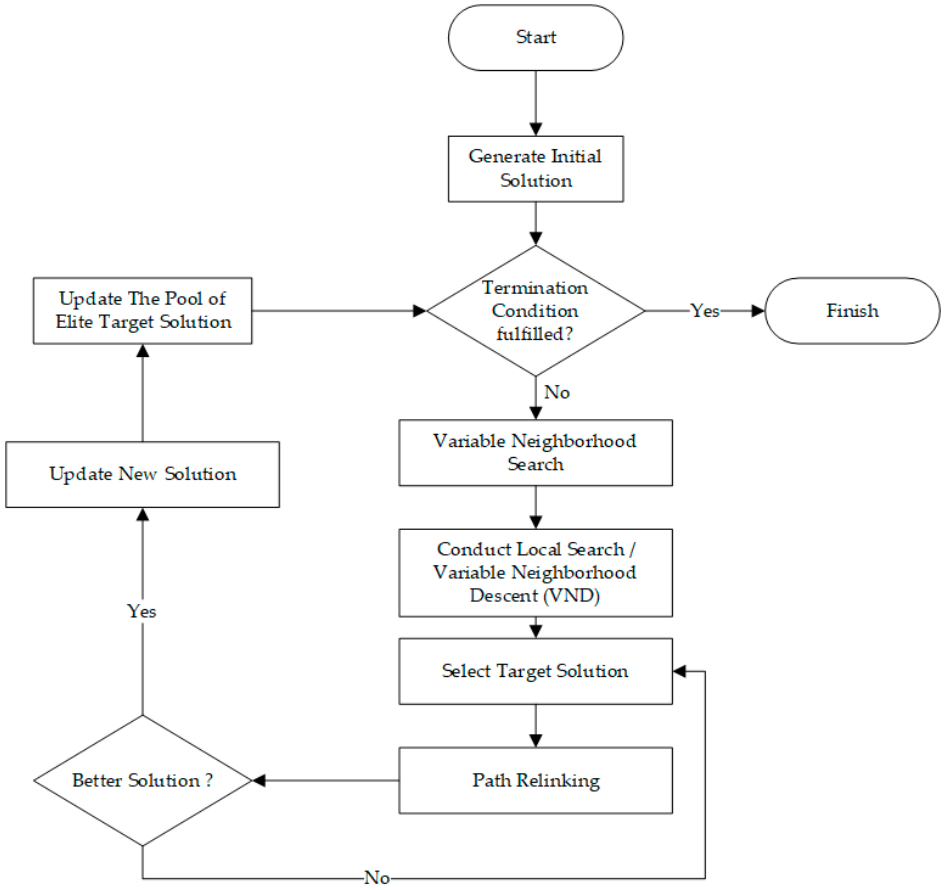


Figure 1. VNSPR framework.

2.2.2. Initialization

The computational result is exemplified by a combination of n customers (1, 2, ..., n), m prospective depots ($n + 1$, $n + 2$, ..., $n + m$), and N_{dummy} zeros. These N_{dummy} zeros are utilized to arbitrarily dismiss the routes to increase the chance for a better solution. The calculation of N_{dummy} number is based on the aggregate of division between demand of customers i and the vehicle capacity Q. This also will ensure that a route will be terminated if the vehicle capacity constraint is violated. During the initialization phase, a number representing a depot will always be the first number in the solution representation. Next, it is followed by a number in {1, 2, ..., n} which represents the next customer to be served. Based on the solution representation, each depot will provide service to the customers located between the depots. Authors attempted not to contravene the time windows of each customer when developing the initial solution. Note that capacity constraints can be violated to search for a better solution. However, the final solution will be made feasible. The detailed algorithm for the initial solution generation can be found from [3,49]. The solution starts with the initially designated depot, then is followed by customers selected to be served by an open depot. Once all customers have been allotted to the first depot based on the depot capacity, the second depot will be chosen and incorporated to the solution representation. This procedure is carried out until each and every customer has been

designated to a depot. Then, N_{dummy} zeros is randomly added. Finally, the closed depot is affixed to the solution. Figure 2 shows the example of the generated initial solution.

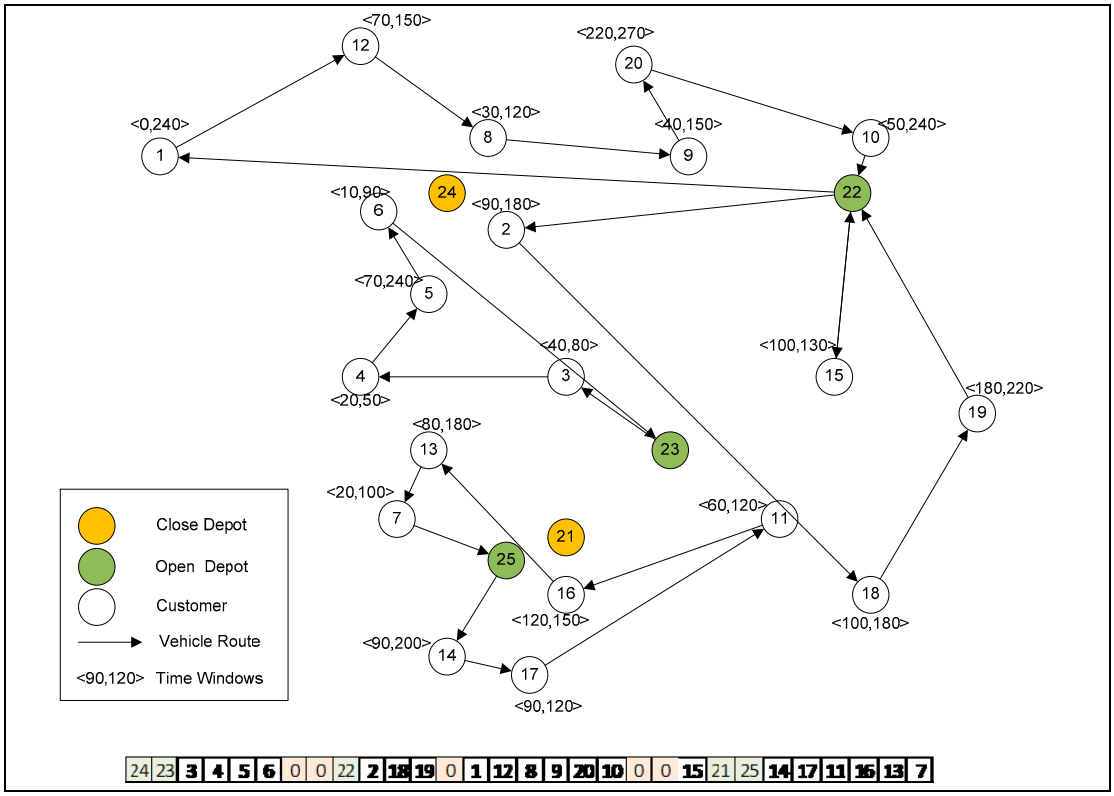


Figure 2. Illustration of example solution representation.

The first depot in this example is 24, and yet because the following number is 23, which is also a depot, Depot 24 is unutilized. Customers 3, 4, 5, and 6 will be served by Depot 23. Based on Figure 2, Depot 22 will have three routes. The representation solution's number 0 functions as the route termination and indicates that the route is rescinded after serving customer 19. The second route will end once its visited customer 10, and the final route from Depot 23 will only serve customer 15. As Depot 21 is unutilized, Depot 25 will be open and serve customers 7, 11, 13, 14, 16, and 17 according to the sequence shown in Figure 2.

2.2.3. Variable Neighborhood Search with Path Relinking Procedure

There are six parameters used for the VNSPR, including k_{max}, $N_{non\text{-}improving}$, m, EX, P and I_{iter}. The first parameters k_{max} is the total neighborhoods incorporated in the shaking procedure. The neigborhood can be varied and pre-selected. The neighborhood variation can be carried out arbitrarily or according to a predetermine sequence. In order to achieve better performance, this algorithm also uses parameter m which is the selected modulus number. This modulus number ensures that the shaking phase will give a better solution each time. Two parameters to stop the algorithm, namely $N_{non\text{-}improving}$ and I_{iter} indicate the maximum number of iterations that can be executed without improving the best objective function value and maximum number of iterations. The algorithm will be terminated if either one of those parameters is fulfilled. A set of elite solutions that consist of candidate target solutions in the pool is equivalent to EX. This solution pool is used in determining the target solution for the path relinking procedure. It is updated during the iteration when a better solution compared to the best-known solution is found. Lastly, P, is the unit penalty cost associated with violation of the time windows constraint.

Initially, a solution X is created with X set as the solution and $obj(X,P)$ set as the objective value. Current solution X is associated with a subset $N(X,P)$, called neighborhood of X. Let $N_k(k = 1, 2, ..., k_{max})$ represent a predefined set of pre-designated neighborhood structures, and $N_k(X,P)$ represents the list of solution in the kth neighborhood. At the beginning, k is equal to 1. For each iteration, an active solution Y is calculated with reference to the kth neighborhood, $N_k(X,P)$. A more intense search is then performed on the solution Y to produce the Y'. The value of $obj(Y',P)$ is then analyzed as a decision to move or not move. If $obj(Y',P)$ is better than $obj(X,P)$, the result X is rationalized, and the procedure is repeated with the same neighborhood N_1. Otherwise, set $k = k + 1$ and repeat the procedures in conjunction with the subsequent neighborhood, N_{k+1}. This process is carried out until $k = k_{max}$. X_{best} will save the best solution observed during the procedure. When k modulus m equals zero, the best solution (X_{best}) is used to renew the current solution, in order to avoid getting stuck at a local optimal solution.

Following that, the PR investigates the route that must be taken to achieve an elite solution X' from a local optimum, X, which was obtained using VNS by choosing random X' from EX. The PR employs a forward strategy to convert the solution X into the guiding solution X' by fixing the broken pairs in X from left to right, based on a non-distance path from X to X'. The procedure searches for a node called I that is present in more than one position in both X and X'. In order to rectify the imbalance, the node I will switch places with the node that was originally occupying the same position in X'.

The time window constraint is treated as hard time windows, where it cannot be violated. However, during the computational procedure, it is allowed to violate the constraints. Every time the solution exceeds or violates the time constraint, it is penalized as P. Finally, the best solution chosen must be feasible, of which the sum of the penalty (P) is equal to zero. The process will be ended once the iteration reaches the I_{iter} or the current best solution, X_{best}, is not improved in $N_{non\text{-}improving}$. The best solution chosen will be a solution with argmin $_{X \in EX}$ obj(X,P) among the EX. The neighborhood structure and the detailed part of the local search will be discussed on the section below.

2.2.4. Neighborhood Structure

The VNSPR algorithm features a five-neighborhoods structure in order to investigate various possibilities regarding the locations of depots and to enhance the way customers are assigned to each depot. In this particular algorithm, the details of the neighborhoods that are used are referred to as insertion, swap, 2-opt, cross-exchange, and Icross-exchange, respectively. The neighborhoods are chosen based on the previous study using VNSPR algorithm conducted by Yu and Maghfiroh [49].

2.2.5. Local Search

The VNSPR frameworks include variable neighborhood descent (VND) as a local search component to enhance the performance of the VNS. It is inspired by a similar approach by Yu and Maghfiroh [49]. The neighborhood was chosen by incorporating the VND method's consecutive deterministic order of neighborhoods [53–55]. Its processes are involved because the neighborhood structures are investigated in the given order, one by one. Unless it finds a local optimum with regard to the neighborhood N_k, the algorithm moves to a neighborhood N_{k+1}. Otherwise, it goes back to the first one, N_1. When k reaches the k'_{max}, the algorithm will be terminated. This k'_{max} is set to a fixed value of five. The effectiveness of VND is equivalent to the order in which neighborhoods are considered and the order in which solutions are explored in each neighborhood. The neighborhoods are ranked according to their increasing cardinality.

2.2.6. Post Optimization: Path Relinking

In order to broaden and deepen the range of the solution pools, a path relinking algorithm was employed. This post optimization holds a pool of the elite solutions EX that has been spawned by VNS. Each solution discovered through a local search is evaluated to

determine whether or not it should be included in EX. The EX will serve as the groundwork for the solutions that are developed. The proposed VNSPR uses the PR operator to examine a path that lies between a local optimum which has been ascertained by VNS and a solution that has been arbitrarily selected from EX.

For a solution X to be considered for incorporation in EX, it must be superior to the solution with least quality in the solution pool. EX is initially populated with solutions that were produced using VNS, and EX is always maintained in an order that is based on $obj(X)$. The Hamming distance, which is equivalent to the number of discrepancies between the nodes at the same position in X and X', is then used to direct travel along the path from X to X'.

The PR operator fixes the broken pairs of X from left to right in order to modify the initial solution X into the guiding solution X'., establishing a route for massive tours with non-increasing distance to X'. The operator used is a forward strategy by swapping solution X to solution X'. The procedure begins by searching for a customer i whose positions in X and X' are distinct from one another. If i is located, then it undergoes permutation with another node in order to correct the disparity. Swapping is performed until $\Delta(X, X')$ is null. Based on the encoded customer sequence described earlier, customers are inserted into service vehicle routes. During this process, we apply two steps sequentially to check if the insertion of each customer is feasible. The post-optimization using path relinking is inspired by by Prins et al. [39]. This study further describes the procedure of path relinking implementation for solving the capacitated location routing problem.

3. Results

The performance of our numerical simulations with the VNSPR algorithm is presented in this section of the paper. The VNSPR algorithm is inscribed in Visual C++ with selected parameter setting k_{max} = 20, m = 5, I_{iter} = 500, $N_{non\text{-improving}}$ = 40, EX = 10, and P = 600.

3.1. Benchmark Instances

Since benchmarks for LRPTW are not available, we derived LRPTW instances from the three most notable LRP instances designed by Barreto [41], Prins et al. [40], and Tuzun and Burke [56]. We generated the time windows and defined the service time based on the best solution SALRP* from Yu et al. [3], for both the customers and depots. The service time for each customer is set to be fixed for all instances. In generating the LRPTW instances, we add a tolerance time for the earliest and latest time for each customer. Due to its fixed tolerance value, the instances generated have different kinds of time window characteristic. Considering that the datasets are produced based on the best solution of LRP, the best-known solution of this problem will be the same as the LRP problem.

3.2. Algorithm Verification

The VNSPR algorithm is initially implemented on all three sets of CLRP benchmark instances so that verification can be performed. The proposed algorithm is then evaluated in comparison to four of the most recently developed heuristics that have been proposed to solve LRP. These heuristics are SALRP [3], GRASP+ ELS [57], modified ant colony algorithm (MACO) [58], and two-phase hybrid heuristic (HH) [59]. Yu et al. [3] reported that these four algorithms produce among the best solutions for LRP. The notations in Table 1 are used for Tables 2–4. The result formatted in bold indicates that it either achieve the same value as the Best Known Solution (BKS) or represent the best result among the results generated by other algorithms for the particular instance.

Table 1. Notation for CLRP instance.

Notation	Description
Prob. ID	problem number
n	quantity of customers
m	quantity of depots
Q	vehicle capacity
BKS	Best Know Solution
Gap %	percentage gap of the solution cost found with respect to BKS
CPU	running time in seconds on the CPU used by each algorithm

Table 2. Results of CLRP for the Barreto's [41] instances.

Prob. ID	n	m	Q	BKS	SALRP		GRASP + ELS		MACO		HH		VNSPR	
					Gap	CPU	Gap	CPU	Gap	CPU	Gap	CPU	Gap	CPU
B1	21	5	6000	**424.9**	**0.00%**	18.3	**0.00%**	0	**0.00%**	5.64	**0.00%**	6	**0.00%**	2.2
B2	22	5	4500	**585.1**	**0.00%**	16.6	**0.00%**	15	**0.00%**	4.69	**0.00%**	9	**0.00%**	3.3
B3	29	5	4500	**512.1**	**0.00%**	23.9	**0.00%**	9	**0.00%**	8.59	**0.00%**	11	**0.00%**	5.2
B4	32	5	8000	**562.2**	**0.00%**	27	**0.00%**	18	**0.00%**	13.05	**0.00%**	40	**0.00%**	9.1
B5	32	5	11,000	**504.3**	**0.00%**	25.1	**0.00%**	34	**0.00%**	9.56	**0.00%**	22	**0.00%**	10.2
B6	36	5	250	**460.4**	**0.00%**	31.7	**0.00%**	0	**0.00%**	13.14	**0.00%**	39	**0.00%**	10.8
B7	50	5	160	**565.6**	**0.00%**	52.8	**0.00%**	8	**0.00%**	29.16	2.62%	45	**0.00%**	15
B8	75	10	140	**844.4**	0.43%	126.8	**0.76%**	86	0.06%	58.72	0.53%	94	**0.00%**	110.3
B9	100	10	200	**833.4**	0.59%	330.8	**0.00%**	127	0.40%	83.92	0.62%	234	**0.00%**	184
B10	12	2	140	**204**	**0.00%**	6.8	-	-	**0.00%**	2.09	-	-	**0.00%**	0.8
B11	55	15	120	**1112.1**	0.06%	112.4	-	-	0.04%	28.81	-	-	**0.00%**	37.4
B12	85	7	160	**1622.5**	**0.00%**	213.1	-	-	0.04%	77.86	-	-	0.63%	120.5
B13	318	4	25,000	**557,275**	1.12%	2806.5	-	-	0.53%	831.34	-	-	0.35%	2189.07
B14	318	4	8000	**670,119**	2.10%	3352.5	-	-	**0.00%**	1986.33	-	-	1.28%	2614.95
B15	27	5	2500	**3062**	**0.00%**	23.3	**0.00%**	35	**0.00%**	8.59	**0.00%**	11	**0.00%**	4.8
B16	134	8	850	**5709**	**0.00%**	522.4	**0.18%**	280	**0.00%**	136.63	3.18%	252	**0.00%**	407.472
B17	88	8	9,000,000	**355.8**	**0.00%**	226.9	**0.00%**	130	**0.00%**	99.53	1.74%	148	**0.00%**	148.4
B18	150	10	8,000,000	**43,919.9**	2.71%	577	**0.10%**	1697	0.48%	166.95	1.50%	456	0.90%	450.06
B19	117	14	150	**12,290.3**	1.17%	323.4	-	-	0.53%	77.45	-	-	2.00%	252.252
Mean gap (%)					0.43%		0.08%		0.11%		0.78%		0.27%	

Table 3. Results of CLRP for the Prins et al.'s [40] instances.

Prob. ID	n	m	Q	BKS	SALRP		GRASP + ELS		MACO		HH		VNSPR	
					Gap	CPU	Gap	CPU	Gap	CPU	Gap	CPU	Gap	CPU
P1	20	5	70	**54,793**	**0.00%**	19.8	**0.00%**	0	**0.00%**	4.08	**0.00%**	3	**0.00%**	3.7
P2	20	5	150	**39,104**	**0.00%**	15	**0.00%**	0	**0.00%**	4.78	**0.00%**	4	**0.00%**	3.8
P3	20	5	70	**48,908**	**0.00%**	19.3	**0.00%**	0	**0.00%**	3.92	0.08%	3	**0.00%**	4.4
P4	20	5	150	**37,542**	**0.00%**	15	**0.00%**	0	**0.00%**	5.45	**0.00%**	4	**0.00%**	3.3
P5	50	5	70	**90,111**	**0.00%**	74.7	**0.00%**	3	**0.00%**	24.47	0.32%	27	**0.00%**	34.3
P6	50	5	150	**63,242**	**0.00%**	57.7	**0.00%**	0	**0.00%**	21.31	1.31%	27	**0.00%**	26
P7	50	5	70	**88,298**	**0.00%**	95	0.39%	11	**0.00%**	24.19	1.18%	23	**0.00%**	37.4
P8	50	5	150	**67,308**	0.05%	58.6	**0.00%**	16	**0.00%**	20.13	1.74%	21	0.10%	25.6
P9	50	5	70	**84,055**	**0.00%**	74.7	**0.00%**	0	**0.00%**	24.66	**0.00%**	23	**0.00%**	40.8
P10	50	5	150	**51,822**	**0.00%**	66.1	**0.00%**	11	**0.00%**	16.83	0.51%	29	**0.00%**	30.2
P11	50	5	70	**86,203**	0.29%	74	**0.00%**	0	**0.00%**	32.78	**0.00%**	66	**0.00%**	36.7
P12	50	5	150	**61,830**	1.41%	58.2	**0.00%**	0	**0.00%**	25.7	**0.00%**	38	**0.00%**	27
P13	100	5	70	**275,419**	0.59%	348.6	0.56%	148	0.29%	116.59	0.28%	157	0.87%	204
P14	100	5	150	**213,615**	1.12%	268.9	1.05%	68	0.33%	134.52	0.60%	136	0.16%	176
P15	100	5	70	**193,671**	0.23%	348.6	0.31%	212	0.40%	237.28	0.49%	145	0.09%	210.3
P16	100	5	150	**157,150**	**0.00%**	211.5	0.14%	125	0.05%	144.33	0.11%	193	0.54%	180.5
P17	100	5	70	**200,079**	0.08%	250.3	0.13%	141	0.48%	178.75	0.50%	163	0.08%	207.9
P18	100	5	150	**152,441**	0.02%	196.7	0.06%	221	0.18%	151.58	0.80%	168	1.47%	164.3
P19	100	10	70	**287,983**	1.06%	270	4.67%	48	1.09%	105.28	0.62%	277	0.19%	147.9
P20	100	10	150	**231,763**	1.06%	202.6	16.32%	186	1.55%	81.81	2.69%	152	1.53%	137.1
P21	100	10	70	**243,590**	0.91%	260.6	0.08%	260	0.69%	122.47	0.89%	92	1.18%	183.705
P22	100	10	150	**203,988**	0.65%	199.3	**0.00%**	139	0.75%	85.03	0.13%	99	0.60%	127.545
P23	100	10	70	**250,882**	**0.00%**	338.1	1.05%	164	1.36%	112.09	1.53%	125	0.87%	168.135
P24	100	10	150	**204,317**	0.34%	240.3	0.38%	203	0.23%	79.7	0.74%	144	0.21%	119.55
P25	200	10	70	**476,778**	0.89%	1428.1	2.03%	1521	0.43%	941.78	**0.00%**	671	0.76%	1130.14
P26	200	10	150	**378,289**	1.40%	1335.8	1.07%	359	0.15%	562.05	**0.00%**	476	0.05%	843.075
P27	200	10	70	**449,849**	0.22%	1795.8	0.54%	112	0.36%	703.66	0.02%	483	0.34%	1055.49
P28	200	10	150	**374,330**	0.63%	1245.1	0.45%	1610	0.17%	403.53	0.17%	530	0.38%	605.295
P29	200	10	70	**472,321**	0.33%	1776	1.28%	1596	0.60%	878.83	**0.00%**	624	0.65%	949.136
P30	200	10	150	**362,817**	0.24%	1326.4	0.65%	591	0.71%	490.92	0.12%	389	0.09%	736.38
Mean gap (%)					0.38%		1.04%		0.33%		0.49%		0.34%	

Table 4. Results of CLRP for the Tuzun and Burke's [56] instances.

Prob. ID	n	m	Q	BKS	SALRP		GRASP + ELS		MACO		HH		VNS-PR	
					Gap	CPU	Gap	CPU	Gap	CPU	Gap	CPU	Gap	CPU
T1	100	10	150	1467.68	0.65%	369.2	0.39%	233	1.50%	71.08	0.79%	152	**0.33%**	105.65
T2	100	20	150	1449.2	1.50%	273.5	**0.00%**	9	0.32%	46.33	2.56%	239	0.14%	114.49
T3	100	10	150	1394.8	0.99%	230.9	0.13%	112	0.93%	60.98	0.89%	120	**0.01%**	79.85
T4	100	20	150	1432.29	**0.00%**	420.4	**0.00%**	114	0.08%	54.05	2.91%	146	**0.00%**	104.13
T5	100	10	150	1167.16	0.86%	348.1	**0.00%**	27	3.50%	79.64	**0.00%**	232	0.03%	102.85
T6	100	20	150	1102.24	0.74%	342.3	**0.00%**	259	**0.00%**	64.53	**0.00%**	224	**0.00%**	119.859
T7	100	10	150	791.66	**0.00%**	359.6	0.05%	5	0.16%	95.38	**0.00%**	201	0.23%	88.95
T8	100	20	150	728.3	0.50%	418.1	**0.00%**	48	**0.00%**	65.14	**0.00%**	254	**0.00%**	120.92
T9	100	10	150	1238.49	**0.00%**	300.3	0.15%	55	2.16%	77.17	**0.00%**	160	**0.00%**	109.4
T10	100	20	150	1245.31	0.16%	427.8	**0.06%**	233	0.93%	50.03	0.47%	237	0.09%	104.48
T11	100	10	150	902.26	**0.00%**	290.5	**0.00%**	249	**0.00%**	60.56	**0.00%**	135	**0.00%**	106.506
T12	100	20	150	1018.29	0.56%	315.6	**0.00%**	196	**0.00%**	68.56	**0.00%**	157	0.20%	111.59
T13	150	10	150	1866.75	4.67%	743.3	4.17%	518	4.21%	226.63	5.09%	485	**3.18%**	186.67
T14	150	20	150	1833.95	3.55%	835.1	1.65%	705	1.05%	100.88	1.23%	298	**0.12%**	185.15
T15	150	10	150	1978.27	4.01%	456.3	0.71%	727	**0.67%**	200.81	1.74%	406	0.94%	205.19
T16	150	20	150	1801.39	**0.00%**	833	1.88%	415	0.61%	140.59	0.09%	302	1.37%	281.55
T17	150	10	150	1445.25	0.56%	749.7	0.59%	103	3.72%	206.06	**0.00%**	449	0.70%	287.17
T18	150	20	150	1441.98	0.94%	827.7	**0.15%**	662	0.32%	162.73	0.70%	493	0.58%	311.74
T19	150	10	150	1204.42	0.15%	752.4	1.28%	459	0.03%	218.19	**0.00%**	270	0.05%	315.8
T20	150	20	150	930.99	0.39%	841.8	1.59%	224	0.08%	149.86	**0.05%**	335	0.21%	327.49
T21	150	10	150	1699.92	1.23%	741.7	0.72%	271	1.42%	226.11	**0.32%**	444	1.19%	302.53
T22	150	20	150	1400.01	1.13%	832.5	0.21%	524	**0.07%**	122.86	1.19%	342	0.20%	255.45
T23	150	10	150	1199.51	1.44%	755.6	1.28%	251	1.48%	240.59	2.94%	526	**0.76%**	308.55
T24	150	20	150	1152.18	0.60%	837	0.33%	375	0.51%	129.94	0.34%	380	**0.07%**	343.22
T25	200	10	150	2265.59	2.58%	1327.9	1.34%	655	1.72%	460.73	**0.00%**	522	0.33%	1334.49
T26	200	20	150	2166.43	4.23%	1454.6	1.71%	432	0.98%	230.83	**0.00%**	603	1.70%	1412.76
T27	200	10	150	2231.46	1.29%	1319.4	0.67%	1566	**0.00%**	427.89	0.80%	527	1.44%	1386.33
T28	200	20	150	2237.81	3.96%	1427.6	1.24%	2192	1.69%	233.89	**0.00%**	558	2.28%	1190.25
T29	200	10	150	2089.77	1.09%	1320.3	0.80%	1521	**0.42%**	569.08	1.54%	522	0.85%	1386.81
T30	200	20	150	1709.56	0.79%	1400.3	4.06%	618	**0.10%**	276.17	2.31%	691	0.84%	1726.62
T31	200	10	150	1466.62	**0.17%**	1299.2	0.52%	514	0.43%	544.3	0.45%	724	0.57%	1481.7
T32	200	20	150	1082.59	0.56%	1429	0.29%	1243	0.46%	316.2	**0.00%**	616	0.11%	1672.26
T33	200	10	150	1970.44	1.20%	1317.5	1.72%	1451	0.42%	386.44	0.73%	542	**0.28%**	1530.21
T34	200	20	150	1918.93	**0.68%**	1411.7	2.37%	1273	2.13%	229.63	2.09%	617	1.90%	1478.13
T35	200	10	150	1771.06	0.45%	1313.8	0.44%	1398	0.67%	405.49	**0.42%**	697	0.51%	1294.5
T36	200	20	150	1390.87	0.40%	1427.4	4.53%	2202	0.13%	268.45	**0.00%**	518	0.52%	1475.13
	Mean gap (%)				1.17%		0.97%		0.91%		0.82%		0.60%	

The performance of the VNSPR is evaluated based on the cost of the solution as well as the percentage difference in the solution that exists between the BKS, the results produced by the VNSPR, and the outcomes produced by other algorithms. The gap is computed for each instance by dividing the difference in value that exists between the proposed algorithm's solution values and the BKS value. Based on the quality of the results, VNSPR has the potential to replace SALRP, GRASP+ ELS, and the two-phase hybrid heuristic (HH). This is measured in terms of the mean gap percentage. The best result for MACO was found while the parameters were being set, whereas the best result for VNSPR is based on a single run, despite the solution quality being a little bit lower. The computational findings demonstrate that VNSPR is capable of successfully resolving CLRP. It is important to note that VNSPR has the smallest average percentage gap out of all of the algorithms in the comparisons; this indicates that the algorithm is efficient and reliable in its ability to solve CLRP.

VNSPR, conversely, has a higher computational effort compared with some algorithms due to the utilization of VND as intensification algorithm. On the other hand, when it comes to a strategic issue that does not need to be resolved on a daily basis, the current computational time is regarded as reasonable. Table 5 shows a summary of the results comparison between the proposed algorithm and other algorithms.

Table 5. Computational results CLRP for all instances compared with other algorithms.

Algorithm	Barreto [41]		Prins et al. [40]		Tuzun and Burke [56]	
	Number of BKS Found	Gap (%)	Number of BKS Found	Gap (%)	Number of BKS Found	Gap (%)
VNSPR	13	0.33	10	0.4	5	0.6
SALRP	12	0.43	11	0.38	3	1.17
SALRP*	18	0.02	27	0.01	21	0.21
HH	7	0.78	8	0.49	13	0.82
MACO	11	0.11	12	0.33	4	0.91

For Barreto's instances, the suggested VNSPR can unravel 14 problems of all of the 19 datasets in these instances. The algorithm, however, is unable to find an optimal solution, particularly for instances B12, B13, B14, B18, and B19, in which aside from instance B12, are large-sized problems with more than 100 customer nodes. The percentage difference over BKS varies between 0.00 and 2.00%, with a mean percentage difference of 0.27%. In terms of mean percentage difference, VNSPR are able to outperform the SALRP algorithm. VNSPR performs better than SALRP and MACO algorithms in terms of quantity of BKSs discovered. For Prins et al. instances, among 30 instances of this dataset, the proposed VNSPR obtains 11 best solutions when compared to the best-known solutions. The percentage difference between BKS and results of the proposed VNSPR spans between 0.00 and 1.53%, with a mean percentage difference is 0.34%. VNSPR's mean percentage difference is smaller than those of SALRP, GRASP + ELS, and HH, although MACO has a slightly lower mean percentage gap of 0.33%. Although VNSPR could not find an optimal solution for a large-sized problem (nodes > 100), the proposed algorithm has a 1.53% smaller maximum gap compared to MACO and still outperforms MACO in large instances.

In the case of Tsuzun and Burke's instances, VNSPR achieves five best-known solutions, with average gaps between them ranging from 0.0% to 3.18%. VNSPR is competitive in terms of solution quality with other algorithms (SALRP, GRASP+ ELS, MACO, and HH), despite the fact that it is unable to discover the best new solutions. This is due to the smallest average percentage gap. In addition, the results demonstrate that the proposed VNSPR methodology has the smallest difference (3.18%) between the smallest and largest gaps. This is the smallest difference that was discovered between any of the algorithms that were taken into consideration for this study.

4. Discussion

Considering that the instances are spawned based on the best solution of LRP, the best-known solution of this problem will be the same as the LRP problem. In this problem, the effectiveness of VNSPR can be evaluated based on the following notations:

Cost	Solution cost achieved by VNSPR algorithm;
Gap %	Percentage difference between the cost of the solution found by the VNSPR algorithm and that of the BKS (the best solution is based on CLRP's solution);
NV	Number of vehicles used.

From a total of 85 instances to solve, VNSPR can find the 21 best solutions. The average gap after adding the time window constraint increases slightly. In addition, total vehicles used also increase in some instances due to a strict time window constraint. Consequently, there is an additional cost. In Barreto's instance, an increase in vehicles used results in an impact to total travel distance while in two other instances, it results in impact to vehicle cost and the total travel distance. The detailed results are shown in Tables 6–8.

Table 6. Computational results for LRP with time windows for Barreto's instances.

Prob. ID	n	m	Q	BKS	NV	VNSPR	NV	Gap (%)
B1	21	5	6000	424.9	4	424.9	4	**0.00%**
B2	22	5	4500	585.1	3	585.1	3	**0.00%**
B3	29	5	4500	512.1	4	512.1	4	**0.00%**
B4	32	5	8000	562.2	3	562.2	3	**0.00%**
B5	32	5	11,000	504.3	3	504.3	3	**0.00%**
B6	36	5	250	460.4	4	460.4	4	**0.00%**
B7	50	5	160	565.6	5	565.6	5	**0.00%**
B8	75	10	140	844.4	11	861.171	11	1.99%
B9	100	10	200	833.4	8	857.837	8	2.93%
B10	12	2	140	204	2	204	2	**0.00%**
B11	55	15	120	1112.10	10	1113.30	10	0.11%
B12	85	7	160	1622.50	11	1628.01	11	0.34%
B13	318	4	25,000	557,275	9	564,409.88	9	1.28%
B14	318	4	8000	670,119	27	678,689	27	1.28%
B15	27	5	2500	3062	4	3062	4	**0.00%**
B16	134	8	850	5709	11	5913.36	10	3.58%
B17	88	8	9,000,000	355.8	7	368.231	6	3.49%
B18	150	10	8,000,000	43,919.90	12	44,855	13	2.13%
B19	117	14	150	12,290.30	6	12,583.10	7	2.38%
		Average			7.57895		7.57895	1.00%

Table 7. Computational results LRP with time windows for Prins et al.'s instances.

Prob. ID	n	m	Q	BKS	NV	VNSPR	NV	Gap (%)
P1	20	5	70	54,793	5	54,793	5	**0.00%**
P2	20	5	150	39,104	3	39,104	3	**0.00%**
P3	20	5	70	48,908	5	48,908	5	**0.00%**
P4	20	5	150	37,542	3	37,542	3	**0.00%**
P5	50	5	70	90,111	12	90,111	12	**0.00%**
P6	50	5	150	63,242	6	63,242	6	**0.00%**
P7	50	5	70	88,298	12	88,298	12	**0.00%**
P8	50	5	150	67,308	6	67,340	6	0.05%
P9	50	5	70	84,055	12	84,055	12	**0.00%**
P10	50	5	150	51,822	6	51,822	6	**0.00%**
P11	50	5	70	86,203	12	86,496	12	0.34%
P12	50	5	150	61,830	6	61,830	6	**0.00%**
P13	100	5	70	275,419	24	277,348	24	0.70%
P14	100	5	150	213,615	11	217,336	11	1.74%
P15	100	5	70	193,671	23	193,993	23	0.17%
P16	100	5	150	157,150	11	160,776	11	2.31%
P17	100	5	70	200,079	23	201,497	24	0.71%
P18	100	5	150	152,441	11	155,638	11	2.10%
P19	100	10	70	287,983	26	294,244	26	2.17%
P20	100	10	150	231,763	12	239,026	12	3.13%
P21	100	10	70	243,590	24	246,464	24	1.18%
P22	100	10	150	203,988	11	208,617	11	2.27%
P23	100	10	70	250,882	24	253,070	24	0.87%
P24	100	10	150	204,317	11	211,926	11	3.72%
P25	200	10	70	477,248	47	494,705	47	3.66%
P26	200	10	150	378,351	22	391,889	22	3.58%
P27	200	10	70	449,849	48	454,117	48	0.95%
P28	200	10	150	374,330	22	394,890	23	5.49%
P29	200	10	70	472,472	47	487,694	47	3.22%
P30	200	10	150	362,817	22	377,792	22	4.13%
		Average			22.3684		22.47368	1.42%

Table 8. Computational results LRP with time windows for Tuzun and Burke's instances.

Prob. ID	n	m	Q	BKS	NV	VNSPR	NV	Gap (%)
T1	100	10	150	1467.68	11	1479.50	11	0.81%
T2	100	20	150	1449.20	11	1456.03	11	0.47%
T3	100	10	150	1394.80	10	1394.93	10	0.01%
T4	100	20	150	1432.29	11	1432.96	11	0.05%
T5	100	10	150	1167.16	11	1170.25	11	0.26%
T6	100	20	150	1102.24	10	1104.37	10	0.19%
T7	100	10	150	791.66	11	794.23	11	0.32%
T8	100	20	150	728.3	11	**728.3**	**11**	**0.00%**
T9	100	10	150	1238.49	11	1249.32	11	0.87%
T10	100	20	150	1245.31	11	1247.86	11	0.20%
T11	100	10	150	902.26	12	**902.26**	**12**	**0.00%**
T12	100	20	150	1018.29	11	1020.73	11	0.24%
T13	150	10	150	1866.75	16	1951.25	16	4.53%
T14	150	20	150	1833.95	16	1882.15	16	2.63%
T15	150	10	150	1978.27	17	2020.79	17	2.15%
T16	150	20	150	1801.39	16	1828.45	16	1.50%
T17	150	10	150	1447.66	16	1456.56	16	0.61%
T18	150	20	150	1441.98	15	1460.56	16	1.29%
T19	150	10	150	1205.09	17	1207.05	17	0.16%
T20	150	20	150	930.99	17	933.89	17	0.31%
T21	150	10	150	1699.92	17	1737.36	17	2.20%
T22	150	20	150	1400.01	16	1405.58	16	0.40%
T23	150	10	150	1199.51	16	1214.89	16	1.28%
T24	150	20	150	1152.18	16	1154.92	16	0.24%
T25	200	10	150	2279.92	21	2329.67	21	2.18%
T26	200	20	150	2185.41	22	2247.71	22	2.85%
T27	200	10	150	2234.78	21	2280.79	21	2.06%
T28	200	20	150	2241.04	21	2263.51	21	1.00%
T29	200	10	150	2089.77	21	2129.20	21	1.89%
T30	200	20	150	1709.56	21	1759.26	20	2.91%
T31	200	10	150	1466.62	20	1482.62	21	1.09%
T32	200	20	150	1084.78	22	1091.28	22	0.60%
T33	200	10	150	1970.44	21	2000.46	21	1.52%
T34	200	20	150	1918.93	21	1972.85	21	2.81%
T35	200	10	150	1771.06	21	1790.42	22	1.09%
T36	200	20	150	1393.16	22	1400.70	22	0.54%
		Average			19.3684		19.4737	1.15%

Table 6 displays the results that were obtained from Barreto's datasets. The results demonstrate that the proposed VNSPR is capable of solving all extensive instances of Barreto's datasets. With differences over BKS ranging from 0.00% to 3.58% and an average relative differences of 1.00%, the proposed VNSPR is capable of obtaining the best solution to 9 out of the 19 problems. In instances 117 × 4 and 150 × 5, VNSPR has a higher number of vehicles used. It can happen because these two instances have narrow time windows, which makes it more difficult to solve.

The same goes for Prins et al.'s datasets. When contrasted to the best solutions that are currently known for all 30 instances of this dataset, the proposed VNSPR obtains 10 of the best solutions. Table 7 presents the information of BKS for Prins et al.'s instances and the computational results of the VNSPR. The gap between the solutions varies from 0.00–5.49% with an average gap of 1.42%. The gap happens due to problem complexity as the consequence of the time window constraint. In order to obtain a feasible solution, a larger number of vehicles used were needed. The additional numbers of vehicles increase vehicle cost and the total cost.

Table 8 presents an additional comparison of the results obtained from the use of various algorithms and proposes VNSPR with regard to the third dataset. The proposed

algorithm can solve 36 instances with an average relative gap of 1.15%, and varies from 0.00% to 4.53%. The number of vehicles used is slightly more than the number reported in the result of BKS. VNSPR has an average vehicle use of 19.47 while BKS has an average vehicle use of 19.36.

Overall, the proposed algorithm is efficient and effective for finding good solutions after adding time window constraints. Compared to CLRP's best-known solutions, the average gap is relatively smaller, and the number of vehicles (*NV*) used is marginally higher. Considering the additional constraint added to the instances, this should be a reasonable result.

5. Conclusions

Both the location and the routing problems are investigated simultaneously in this study, i.e., locations depot and assignment of customers with consideration of time windows, depot capacity, and vehicle capacity. This study considers the LRPTW problem as the extension of the LRP problem. We developed a new hybrid method to solve CLRP and LRPTW. Our approach utilizes greedy heuristic to assign open depots and customer selection using a variable neighborhood search for route improvement and path relinking as the intensification method.

An algorithm named variable neighborhood search with path relinking (VNSPR) can effectively solve large-scale instances of CLRP and LRPTW problems. In order to validate the algorithm's performance, the algorithm was applied to three well-established LRP instances. Furthermore, 85 generated instances of LRPTW were tested. Computational results show that the proposed method can solve NP-hard LRPTW problems with good solutions. The results also demonstrate the efficacy and efficiency of the proposed algorithm, as they indicate that results produced are approximately the same as the best-known results. Moreover, VNSPR obtains promising results on almost all of the instances that were tested, which demonstrates the algorithm's robustness.

The proposed VNSPR can be embedded into any metaheuristic approach, such as simulated annealing, to tackle the deterministic phase within the VNS in order to obtain a better solution. It has been observed that LRPTW instances with narrow time windows are harder to solve due to its difficulty in finding a feasible solution. We can utilize time window heuristics as the initial solution to tackle this problem.

Author Contributions: Conceptualization, M.F.N.M. and V.F.Y.; methodology, M.F.N.M. and V.F.Y.; software, M.F.N.M. and A.A.N.P.R.; validation, M.F.N.M., A.A.N.P.R. and V.F.Y.; formal analysis, M.F.N.M.; investigation M.F.N.M., A.A.N.P.R. and V.F.Y.; resources, M.F.N.M., A.A.N.P.R. and V.F.Y.; data curation, M.F.N.M., A.A.N.P.R. and V.F.Y.; writing—original draft preparation, M.F.N.M., writing—review and editing, V.F.Y.; visualization, M.F.N.M., A.A.N.P.R. and V.F.Y.; supervision, V.F.Y. and B.N.A.; project administration, A.A.N.P.R.; funding acquisition, B.N.A. All authors have read and agreed to the published version of the manuscript.

Funding: Funding is provided by LPPM Institut Teknologi Kalimantan research grant 2022.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments: The research of the second author is partially supported by the National Science and Technology Council, Taiwan, under Grant MOST 111-2410-H-011-020-MY3 and the Center for Cyber-Physical System Innovation from the Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education in Taiwan.

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

References

1. Belenguer, J.-M.; Benavent, E.; Prins, C.; Prodhon, C.; Wolfler Calvo, R. A Branch-and-Cut Method for the Capacitated Location-Routing Problem. *Comput. Oper. Res.* **2011**, *38*, 931–941. [CrossRef]
2. Ponboon, S.; Qureshi, A.G.; Taniguchi, E. Branch-and-Price Algorithm for the Location-Routing Problem with Time Windows. *Transp. Res. Part E Logist. Transp. Rev.* **2016**, *86*, 1–19. [CrossRef]
3. Yu, V.F.; Lin, S.-W.; Lee, W.; Ting, C.-J. A Simulated Annealing Heuristic for the Capacitated Location Routing Problem. *Comput. Ind. Eng.* **2010**, *58*, 288–299. [CrossRef]
4. Karaoglan, I.; Altiparmak, F.; Kara, I.; Dengiz, B. The Location-Routing Problem with Simultaneous Pickup and Delivery: Formulations and a Heuristic Approach. *Omega* **2012**, *40*, 465–477. [CrossRef]
5. Zhou, L.; Wang, X.; Ni, L.; Lin, Y. Location-Routing Problem with Simultaneous Home Delivery and Customer's Pickup for City Distribution of Online Shopping Purchases. *Sustainability* **2016**, *8*, 828. [CrossRef]
6. Wang, M.; Zhang, C.; Bell, M.G.H.; Miao, L. A Branch-and-Price Algorithm for Location-Routing Problems with Pick-up Stations in the Last-Mile Distribution System. *Eur. J. Oper. Res.* **2022**, *303*, 1258–1276. [CrossRef]
7. Tayebi Araghi, M.E.; Tavakkoli-Moghaddam, R.; Jolai, F.; Hadji Molana, S.M. A Green Multi-Facilities Open Location-Routing Problem with Planar Facility Locations and Uncertain Customer. *J. Clean. Prod.* **2021**, *282*, 124343. [CrossRef]
8. Nucamendi-Guillén, S.; Gómez Padilla, A.; Olivares-Benitez, E.; Moreno-Vega, J.M. The Multi-Depot Open Location Routing Problem with a Heterogeneous Fixed Fleet. *Expert Syst. Appl.* **2021**, *165*, 113846. [CrossRef]
9. Yu, V.F.; Lin, S.-Y. A Simulated Annealing Heuristic for the Open Location-Routing Problem. *Comput. Oper. Res.* **2015**, *62*, 184–196. [CrossRef]
10. Yu, X.; Zhou, Y.; Liu, X.-F. The Two-Echelon Multi-Objective Location Routing Problem Inspired by Realistic Waste Collection Applications: The Composable Model and a Metaheuristic Algorithm. *Appl. Soft Comput.* **2020**, *94*, 106477. [CrossRef]
11. Fallahtafti, A.; Ardjmand, E.; Young, W.A.; Weckman, G.R. A Multi-Objective Two-Echelon Location-Routing Problem for Cash Logistics: A Metaheuristic Approach. *Appl. Soft Comput.* **2021**, *111*, 107685. [CrossRef]
12. Rabbani, M.; Heidari, R.; Farrokhi-Asl, H.; Rahimi, N. Using Metaheuristic Algorithms to Solve a Multi-Objective Industrial Hazardous Waste Location-Routing Problem Considering Incompatible Waste Types. *J. Clean. Prod.* **2018**, *170*, 227–241. [CrossRef]
13. Tavakkoli-Moghaddam, R.; Makui, A.; Mazloomi, Z. A New Integrated Mathematical Model for a Bi-Objective Multi-Depot Location-Routing Problem Solved by a Multi-Objective Scatter Search Algorithm. *J. Manuf. Syst.* **2010**, *29*, 111–119. [CrossRef]
14. Toro, E.M.; Franco, J.F.; Echeverri, M.G.; Guimarães, F.G. A Multi-Objective Model for the Green Capacitated Location-Routing Problem Considering Environmental Impact. *Comput. Ind. Eng.* **2017**, *110*, 114–125. [CrossRef]
15. Cheng, C.; Zhu, R.; Costa, A.M.; Thompson, R.G.; Huang, X. Multi-Period Two-Echelon Location Routing Problem for Disaster Waste Clean-Up. *Transp. A Transp. Sci.* **2022**, *18*, 1053–1083. [CrossRef]
16. Ben Mohamed, I.; Klibi, W.; Sadykov, R.; Şen, H.; Vanderbeck, F. The Two-Echelon Stochastic Multi-Period Capacitated Location-Routing Problem. *Eur. J. Oper. Res.* **2022**, *306*, 645–667. [CrossRef]
17. Dai, Z.; Aqlan, F.; Gao, K.; Zhou, Y. A Two-Phase Method for Multi-Echelon Location-Routing Problems in Supply Chains. *Expert Syst. Appl.* **2019**, *115*, 618–634. [CrossRef]
18. Xu, X.; Liu, W.; Jiang, M.; Lin, Z. A Multi-Cycle and Multi-Echelon Location-Routing Problem for Integrated Reverse Logistics. *Ind. Manag. Data Syst.* **2022**, *122*, 2237–2260. [CrossRef]
19. Wang, Y.; Wang, N.; Han, P. Maritime Location Inventory Routing Problem for Island Supply Chain Network under Periodic Freight Demand. *Comput. Oper. Res.* **2023**, *149*, 106042. [CrossRef]
20. Yavari, M.; Enjavi, H.; Geraeli, M. Demand Management to Cope with Routes Disruptions in Location-Inventory-Routing Problem for Perishable Products. *Res. Transp. Bus. Manag.* **2020**, *37*, 100552. [CrossRef]
21. Ahmadi Javid, A.; Azad, N. Incorporating Location, Routing and Inventory Decisions in Supply Chain Network Design. *Transp. Res. Part E Logist. Transp. Rev.* **2010**, *46*, 582–597. [CrossRef]
22. Guerrero, W.J.; Prodhon, C.; Velasco, N.; Amaya, C.A. Hybrid Heuristic for the Inventory Location-Routing Problem with Deterministic Demand. *Int. J. Prod. Econ.* **2013**, *146*, 359–370. [CrossRef]
23. Partovi, F.; Seifbarghy, M.; Esmaeili, M. Revised Solution Technique for a Bi-Level Location-Inventory-Routing Problem under Uncertainty of Demand and Perishability of Products. *Appl. Soft Comput.* **2022**, 109899. [CrossRef]
24. Biuki, M.; Kazemi, A.; Alinezhad, A. An Integrated Location-Routing-Inventory Model for Sustainable Design of a Perishable Products Supply Chain Network. *J. Clean. Prod.* **2020**, *260*, 120842. [CrossRef]
25. Rafie-Majd, Z.; Pasandideh, S.H.R.; Naderi, B. Modelling and Solving the Integrated Inventory-Location-Routing Problem in a Multi-Period and Multi-Perishable Product Supply Chain with Uncertainty: Lagrangian Relaxation Algorithm. *Comput. Chem. Eng.* **2018**, *109*, 9–22. [CrossRef]
26. Zabihian-Bisheh, A.; Pourrezaie-Khaligh, P.; Farrokhi-Asl, H.; Lim, S.F.W.; Rabbani, M.; Khamechian, M. Multi-Depot Green Capacitated Location Routing Problem Considering Uncertainty and in-Facility Queuing. *Clean. Waste Syst.* **2022**, *2*, 100011. [CrossRef]
27. Fazayeli, S.; Eydi, A.; Kamalabadi, I.N. Location-Routing Problem in Multimodal Transportation Network with Time Windows and Fuzzy Demands: Presenting a Two-Part Genetic Algorithm. *Comput. Ind. Eng.* **2018**, *119*, 233–246. [CrossRef]
28. Fazel Zarandi, M.H.; Hemmati, A.; Davari, S.; Burhan Turksen, I. Capacitated Location-Routing Problem with Time Windows under Uncertainty. *Knowl. -Based Syst.* **2013**, *37*, 480–489. [CrossRef]

29. Koç, Ç.; Bektaş, T.; Jabali, O.; Laporte, G. The Fleet Size and Mix Location-Routing Problem with Time Windows: Formulations and a Heuristic Algorithm. *Eur. J. Oper. Res.* **2016**, *248*, 33–51. [CrossRef]
30. Dharmika, N.A.; Maruti, S.C.; Sakti, S.; Asih, A.M.S. Multi-Objective Location Routing Problem with Time Windows for Cost Minimization and Customer Service Level Maximization. In Proceedings of the 2020 6th International Conference on Science and Technology (ICST), Odessa, Ukraine, 24–26 September 2020; Volume 1, pp. 1–6.
31. Wei, X.; Qiu, H.; Wang, D.; Duan, J.; Wang, Y.; Cheng, T.C.E. An Integrated Location-Routing Problem with Post-Disaster Relief Distribution. *Comput. Ind. Eng.* **2020**, *147*, 106632. [CrossRef]
32. Çetinkaya, C.; Gökçen, H.; Karaoğlan, İ. The Location Routing Problem with Arc Time Windows for Terror Regions: A Mixed Integer Formulation. *J. Ind. Prod. Eng.* **2018**, *35*, 309–318. [CrossRef]
33. Schiffer, M.; Walther, G. The Electric Location Routing Problem with Time Windows and Partial Recharging. *Eur. J. Oper. Res.* **2017**, *260*, 995–1013. [CrossRef]
34. Navazi, F.; Sedaghat, A.; Tavakkoli-Moghaddam, R. A New Sustainable Location-Routing Problem with Simultaneous Pickup and Delivery by Two-Compartment Vehicles for a Perishable Product Considering Circular Economy. *IFAC-PapersOnLine* **2019**, *52*, 790–795. [CrossRef]
35. Zhalechian, M.; Tavakkoli-Moghaddam, R.; Zahiri, B.; Mohammadi, M. Sustainable Design of a Closed-Loop Location-Routing-Inventory Supply Chain Network under Mixed Uncertainty. *Transp. Res. Part E Logist. Transp. Rev.* **2016**, *89*, 182–214. [CrossRef]
36. Tasouji Hassanpour, S.; Ke, G.Y.; Tulett, D.M. A Time-Dependent Location-Routing Problem of Hazardous Material Transportation with Edge Unavailability and Time Window. *J. Clean. Prod.* **2021**, *322*, 128951. [CrossRef]
37. Shang, C.; Ma, L.; Liu, Y.; Sun, S. The Sorted-Waste Capacitated Location Routing Problem with Queuing Time: A Cross-Entropy and Simulated-Annealing-Based Hyper-Heuristic Algorithm. *Expert Syst. Appl.* **2022**, *201*, 117077. [CrossRef]
38. Gündüz, H.I. *The Single-Stage Location-Routing Problem with Time Windows BT—Computational Logistics*; Böse, J.W., Hu, H., Jahn, C., Shi, X., Stahlbock, R., Voß, S., Eds.; Springer Berlin Heidelberg: Berlin/Heidelberg, Germany, 2011; pp. 44–58.
39. Prins, C.; Prodhon, C.; Calvo, R.W. Solving the Capacitated Location-Routing Problem by a GRASP Complemented by a Learning Process and a Path Relinking. *4OR* **2006**, *4*, 221–238. [CrossRef]
40. Prodhon, C. Solving the Capacitated Location-Routing Problem. *4OR* **2007**, *5*, 339–342. [CrossRef]
41. Barreto, S.S. *Análise e Modelização de Problemas de Localização-Distribuição*; University of Aveiro: Aveiro, Portugal, 2004.
42. Mladenović, N.; Hansen, P. Variable Neighborhood Search. *Comput. Oper. Res.* **1997**, *24*, 1097–1100. [CrossRef]
43. Polacek, M.; Hartl, R.F.; Doerner, K.; Reimann, M. A Variable Neighborhood Search for the Multi Depot Vehicle Routing Problem with Time Windows. *J. Heuristics* **2004**, *10*, 613–627. [CrossRef]
44. Bräysy, O. A Reactive Variable Neighborhood Search for the Vehicle-Routing Problem with Time Windows. *INFORMS J. Comput.* **2003**, *15*, 347–368. [CrossRef]
45. Dhahri, A.; Mjirda, A.; Zidi, K.; Ghedira, K. A VNS-Based Heuristic for Solving the Vehicle Routing Problem with Time Windows and Vehicle Preventive Maintenance Constraints. *Procedia Comput. Sci.* **2016**, *80*, 1212–1222. [CrossRef]
46. Ronconi, D.P.; Manguino, J.L.V. GRASP and VNS Approaches for a Vehicle Routing Problem with Step Cost Functions. *Ann. Oper. Res.* **2022**. [CrossRef]
47. Wu, Y.; Huang, M.; Yu, J.; Wang, Q. A VNS Solution and Objectives Tradeoffs for Real-Life Large-Scale Vehicle Routing Problems. *Int. J. Ind. Syst. Eng.* **2021**, *39*, 565–590. [CrossRef]
48. Redi, A.A.N.P.; Maghfiroh, M.F.N.; Yu, V.F. An Improved Variable Neighborhood Search for the Open Vehicle Routing Problem with Time Windows. In Proceedings of the 2013 IEEE International Conference on Industrial Engineering and Engineering Management, Bangkok, Thailand, 10–13 December 2013; pp. 1641–1645.
49. Yu, V.F.; Maghfiroh, M.F.N. A Variable Neighborhood Search with Path-Relinking for the Capacitated Location Routing Problem. *J. Ind. Prod. Eng.* **2014**, *31*, 163–176. [CrossRef]
50. Karakostas, P.; Sifaleras, A.; Georgiadis, M.C. A General Variable Neighborhood Search-Based Solution Approach for the Location-Inventory-Routing Problem with Distribution Outsourcing. *Comput. Chem. Eng.* **2019**, *126*, 263–279. [CrossRef]
51. Schwengerer, M.; Pirkwieser, S.; Raidl, G.R. *A Variable Neighborhood Search Approach for the Two-Echelon Location-Routing Problem BT—Evolutionary Computation in Combinatorial Optimization*; Hao, J.-K., Middendorf, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 13–24.
52. Pirkwieser, S.; Raidl, G.R. *Variable Neighborhood Search Coupled with ILP-Based Very Large Neighborhood Searches for the (Periodic) Location-Routing Problem BT—Hybrid Metaheuristics*; Blesa, M.J., Blum, C., Raidl, G., Roli, A., Sampels, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 174–189.
53. Harzi, M.; Krichen, S. Variable Neighborhood Descent for Solving the Vehicle Routing Problem with Time Windows. *Electron. Notes Discret. Math.* **2017**, *58*, 175–182. [CrossRef]
54. Chen, P.; Huang, H.; Dong, X.-Y. Iterated Variable Neighborhood Descent Algorithm for the Capacitated Vehicle Routing Problem. *Expert Syst. Appl.* **2010**, *37*, 1620–1627. [CrossRef]
55. Rousseau, L.-M.; Gendreau, M.; Pesant, G. Using Constraint-Based Operators to Solve the Vehicle Routing Problem with Time Windows. *J. Heuristics* **2002**, *8*, 43–58. [CrossRef]
56. Tuzun, D.; Burke, L.I. A Two-Phase Tabu Search Approach to the Location Routing Problem. *Eur. J. Oper. Res.* **1999**, *116*, 87–99. [CrossRef]

57. Duhamel, C.; Lacomme, P.; Prins, C.; Prodhon, C. A GRASP×ELS Approach for the Capacitated Location-Routing Problem. *Comput. Oper. Res.* **2010**, *37*, 1912–1923. [CrossRef]
58. Ting, C.-J.; Chen, C.-H. A Multiple Ant Colony Optimization Algorithm for the Capacitated Location Routing Problem. *Int. J. Prod. Econ.* **2013**, *141*, 34–44. [CrossRef]
59. Escobar, J.W.; Linfati, R.; Toth, P. A Two-Phase Hybrid Heuristic Algorithm for the Capacitated Location-Routing Problem. *Comput. Oper. Res.* **2013**, *40*, 70–79. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.