



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

ARCHIVIO ISTITUZIONALE DELLA RICERCA

Alma Mater Studiorum Università di Bologna Archivio istituzionale della ricerca

Multi-compartment waste collection vehicle routing problem with bin washer

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:

Amine Masmoudi, M., Baldacci, R., Mancini, S., Kuo, Y.-H. (2024). Multi-compartment waste collection vehicle routing problem with bin washer. *TRANSPORTATION RESEARCH PART E-LOGISTICS AND TRANSPORTATION REVIEW*, 189, 1-20 [10.1016/j.tre.2024.103681].

Availability:

This version is available at: <https://hdl.handle.net/11585/1006576> since: 2025-05-12

Published:

DOI: <http://doi.org/10.1016/j.tre.2024.103681>

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>).
When citing, please refer to the published version.

(Article begins on next page)

Multi-Compartment Waste Collection Vehicle Routing Problem with Bin Washer

Abstract

This paper introduces a new variant of the Multi-Compartment Waste Collection Problem (MC-WCP), which we call the MC-WCP with Bin Washer (MC-WCP-BW). The problem involves a fleet of compressed refuse collection trucks equipped with a bin washer, which compresses each waste type into the corresponding compartment of a vehicle. The vehicles can also wash the bins. Separation sites and water refilling stations are considered in the problem. A subset of the bins must be washed when emptied, which is not mandatory for the others. The problem is modeled using a mixed-integer linear programming formulation incorporating multiple visits to separation sites and water refilling stations. An efficient Hybrid Variable Neighborhood Search (HVNS) algorithm is developed and evaluated on a set of instances from the literature and newly generated instances. Computational experiments show that our proposed algorithm can identify solutions of better quality in a shorter computational time, as compared with the current state-of-the-art algorithms. The potential benefits of the compaction operation under several scenarios are also analyzed via extensive analyses.

Keywords: Transportation, multi-compartment vehicle routing problem, waste collection, hybrid metaheuristics, bin washer

1. Introduction

Waste management is a value-adding aspect of sustainable living and economy in many countries. Due to urbanization and growing e-commerce activities, significant volumes of waste (e.g., paper, metal, plastics, and organics) derived from human activities and production processes must be collected, disposed of, and processed sustainably. Applications for waste collection span a wide range of successful real-world practices in various countries. For example, researchers have investigated such promising applications in Spain (Molina et al., 2019), China (Zhang et al., 2020), and Portugal (Oliveira et al., 2015; Ramos et al., 2018). As discussed in the literature, governments, municipalities, and relevant organizations have formulated impactful solutions for sustainable waste collection. The main finding from these studies is the deployment of suitable resources in a more efficient way.

Determining the most efficient set of routes for collecting different types of waste is known as Multi Compartment Waste Collection Vehicle Routing Problem (MC-WCVRP), which has been proven to be NP-hard (Muyldermans and Pang, 2010). Compared with traditional multi-compartment vehicle routing problems (e.g., Chen et al., 2020), MC-WCVRP is more challenging as further constraints related to waste handling must be imposed. Although several variants of MC-WCVRP have been proposed in the

literature, more complex models are needed to capture all the features from reality. Our study aims to propose new concepts and ideas that would enhance the practicality of the proposed approach.

In this paper, we consider the following problem setting: *i*) a fleet of Compressed Refuse Vehicles equipped with Bin Washer (CRVBW) is to be deployed; *ii*) two different sets of stations, including separation and water refilling stations, are considered, and *iii*) the concept of en-route washing bins is introduced.

In most MC-WCVRP studies, only collection services are performed by a fleet of vehicles to empty bins (e.g., [Rabbani et al., 2016, 2017](#); [Farrokhi-Asl et al., 2017](#); [Zbib and Laporte, 2020](#)). However, collecting waste from the bins can lead to bad smells and hygiene issues ([Feng et al., 2022](#)). Therefore, it is necessary to clean and disinfect them regularly. In practice, the washing of bins process is generally done by a specialized fleet of vehicles to ensure the hygiene of bins, such as the bin-cleaning vehicle Hyva ([Hyva company](#)). Using these vehicles can be costly for any waste company due to their high operational costs. In addition, using different vehicles can cause a conflict between emptying and washing operations. Two separate planning problems should be taken into consideration. Delaying the cleaning of bins after the emptying operation may create a health hazard in populated urban areas ([UNEP](#)). Thus, an effective washing of the bins plan must take place directly after emptying the bins. To overcome this challenge, as well as the concerns related to insufficient vehicles, limited operational times, and high costs, the waste collection industry needs to seek innovative and environmental solutions that use CRVBW. These vehicles can collect and compact waste and wash bins simultaneously. In other applications of refuse vehicles for waste collection, we refer to the OBW Refuse Compactors ([Orakçi company](#)), Rafco Refuse Vehicles ([Rafco company](#)), and Fulongma Vehicles ([Fulongma company](#)). Even though there are promising applications, CRVBW in the context of MC-WCVRP has not been previously studied. From an economic point of view, these vehicles can reduce the number of visits to landfill by collecting more bins during the working day. From the waste management point of view, compacting waste takes less space in a landfill facility, where the space is usually limited.

The second challenge is that the refuse vehicle needs to be emptied when one or more compartments are full ([Mofid-Nakhaee and Barzinpour, 2019](#); [Erdem, 2022](#)). The capacity of waste collection vehicles generally cannot accommodate waste from the entire collection area. Their capacity can be saturated by very few bins if they are full. Therefore, several visits to separation sites may be required on the route. In other words, each vehicle performs a single trip starting and ending at the depot but can visit separation sites multiple times. This paper considers CRVBW to collect waste in separate compartments and assumes additional nodes as the intermediate facilities since they transfer the collected wastes to the other treatment facilities, including recycling facilities far from cities ([Mofid-Nakhaee and Barzinpour, 2019](#)).

Such a practice of waste collection with the use of bin washers is already in operation. However, some bins can contain waste that is not appropriate for their collection. Thus, waste must be checked before it is put in the right compartment. These checks can be carried out in different areas of the same facility.

Thus, after collecting the waste, municipalities need to verify the waste and treat them in the treatment facilities before transporting it to the recycling facilities. The separation of recyclable waste (e.g., paper, metal, plastic) from different bins is done at separation sites (Bing et al., 2016). After separation, the waste is sent to a sorting site, where further sorting of the material by colour and/or composition is conducted. The waste is then transported to specialized treatment facilities to be re-melted or transformed for recycling. The separation and sorting processes might occur in different stages for different types of waste (Bing et al., 2016). In addition, CRVBWs may need frequent water filling from a water station (Khajouei et al., 2020) because they have a relatively short tank capacity, around 120 bins that may vary depending on tank water capacity (Orakçi company; ProCompactor company). Thus, a vehicle can fill up its water tank and then be able to perform cleaning operations along the route. In our study, two types of stations are considered; one dedicated as a separation site and the other as a water refilling station.

Finally, we consider the obligation to clean some bins while others may not. For example, bins located in a city center area that are usually densely populated must be cleaned daily, while bins in more peripheral zones can be cleaned less frequently. However, to provide a certain service level, we allow a limited number of bins not to be washed. This provides the company with a margin of tolerance to re-optimize its routes.

Figure 1 presents an example of the MC-WCP-BW involving two vehicles (routes) and 13 collection points. Each collection point (i) contains four different bins that must be emptied and washed (if it is located in the city center area) if necessary, two water refilling stations (w), three separation sites (s), and a value attributed to each vehicle shows the amount of collected waste in each compartment when the vehicle arrives at the node of each collection point (i). In this example, the capacity load of each compartment is set to 100 Kg. In this example, there are four compartments in each vehicle, where each compartment is associated with a single waste type (plastic, paper, metal, or glass). Moreover, the amount of water in the tank when the vehicle washes the bins of a collection point (i) is shown at the node. In this example, the capacity of the water tank of the vehicle is set to 100L.

Vehicle V1 compacts the waste of the first compartment (in blue) at collection point node 2 after loading all different waste types of collection points 6, 9, and 2 in order to be able to collect the waste of collection point 1. In addition, the vehicle visits a water refilling station ($w=1$) in order to be able to wash the bins of collection point 5 since it is located in the city center area and must be washed and the available water in the vehicle tank after washing collection point bins 1 is not enough to wash the bins of collection point 5. After servicing collection point 5, the vehicle visits a separation site ($s=1$) to empty all compartment wastes (that reset to zero) to continue emptying collection point bins 13 and 3. Since the compartments still contain waste and must be returned to the depot, the vehicle then visits the nearest separation site ($s=1$) again to empty all the waste before returning to the depot.

Vehicle V2 collects waste from collection points 11, 8, and 12 before visiting a separation site ($s=2$) without compacting any waste type during the service of these collection points. The vehicle then

continues emptying and washing collection point bins 4, 10, and 7 without visiting any water refilling station.

From Figure 1, a separation site station can be visited many times by the same vehicle, as in the case of separation site node 1 ($s=1$). A water refilling station may not necessarily be visited at all, as in the case of water refilling station 2 ($w=2$). In addition, some bins located outside the city center area do not need to be washed, as in the cases of collection points 11, 8, and 12.

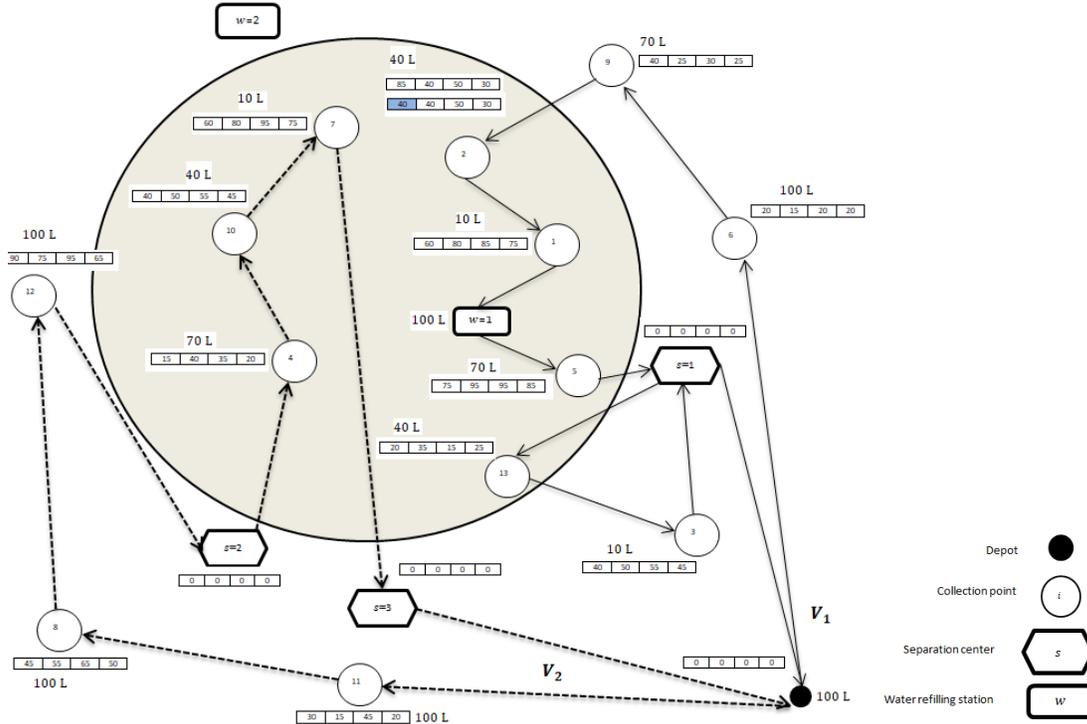


Fig. 1. Illustrative example of the MC-WCP-BW

Our research contributes to both the academic community and practice. From an academic standpoint, our work is distinguished in the following ways:

- i) Our study introduces a new approach for the MC-WCVRP that addresses two key challenges faced in the waste management industry. Firstly, we introduce the use of CRVBW, which can simultaneously collect and compact waste and wash bins. This reduces the number of visits to the landfill, saves space, and lowers operational costs. Secondly, we consider different types of stations, including separation and water refilling stations, to model a more realistic waste collection scenario. We account for the need to visit separation sites multiple times due to the limited capacity of waste collection vehicles and the compaction operation. Our proposed approach also allows for certain bins to be cleaned while others may not, depending on factors such as location or usage. This adds a level of flexibility to the waste collection process and helps manage limited resources more

effectively. Overall, our paper provides innovative and environmentally sustainable solutions for waste collection that have not been previously studied in the context of MC-WCVRP.

- ii)* We present an improved formulation of the MC-WCP-BW problem. The stations to visit are not explicitly represented, as in the study of [Bruglieri et al. \(2019\)](#). A graph consists of bins and depots, connecting each pair of nodes with multiple arcs. Each arc (i, j) of the graph captures a feasible sequence of consecutive bin waste separations and/or water refilling without using copies or dummy vertices.
- iii)* We propose an HVNS metaheuristic algorithm for the MC-WCP-BW. We develop several diversification and intensification mechanisms to improve the efficiency of the Variable Neighborhood Search (VNS) algorithm. Our approach incorporates multiple established metaheuristic methodologies, including the crossover operator of the Genetic Algorithm (GA), adaptive temperature reduction of the Simulated Annealing (SA) algorithm, and an adaptive intelligent mechanism for diversification and intensification. The algorithm includes compaction operations and visiting separation/water refueling stations, which have been shown to make significant contributions.
- iv)* We present evidence of the efficacy of our HVNS and demonstrate that our algorithm outperforms current state-of-the-art algorithms in terms of computational times. In addition, we demonstrate the benefits of using several feature components in the standard VNS. Furthermore, we provide managerial insights and sensitivity analyses regarding the key parameters of the problem, specifically the benefits associated with performing compaction operations to minimize the distance required for separation by collecting more bins. We emphasize the managerial insight of these concepts, such as the impact of capacity variation in the water tank on solution quality and how using different city center area dimensions significantly impacts the quality of the solution, specifically the number of bins that need to be washed.

Our research can provide valuable guidance to route planners and managers in optimizing their waste transportation activities through the use of innovative waste collection vehicles.

The remainder of this paper is organized as follows. An overview of the recent literature is provided in [Section 2](#). A brief description of the problem is given in [Section 3](#), whereas [Section 4](#) discusses our proposed algorithm. [Section 5](#) presents the computational results. [Section 6](#) provides several managerial insights, while conclusions and some suggested future research directions are outlined in [Section 7](#).

2. Literature review

We first review single- and multiple-compartment waste collection and then discuss existing studies in various fields relevant to driving time reduction.

The waste collection with a single compartment is generally used to collect different types of waste in the same compartment and is known as Waste Collection Vehicle Routing Problem (WCVRP). The

concept of WCVRP was first introduced by [Kim et al. \(2006\)](#). The authors considered a WCVRP with time windows, multiple trips, and a lunch break for drivers. Later, [Benjamin and Beasley \(2010\)](#) studied the Waste Collection Vehicle Routing Problem with Time Windows (WCVRPTW), which considers multiple disposal facilities, time windows, and driver rest periods. The authors proposed a VNS algorithm to solve the problem effectively. The algorithm was run on very large instances with up to 2,000 customers and 19 landfill facilities. [Louati et al. \(2016\)](#) introduced a WCVRPTW that takes into account time windows, multiple transfer stations, gather sites, and heterogeneous vehicles operating within specific time windows. The authors proposed a MILP formulation and applied it on a real-life waste collection application in Vietnam. [Tirkolaee et al. \(2019\)](#) proposed a Simulated Annealing (SA) algorithm to solve the multi-trip VRP arising as an urban waste collection problem. [Rabbani et al. \(2018\)](#) considered a mixed homogeneous fleet composed of internal and external refuse vehicles. Different types of waste that require compatible treatment technology are considered, in contrast to our problem, in which all different waste types are emptied in a single disposal facility. In another recent research, [Masmoudi et al. \(2022b\)](#) investigated the utilization of a homogeneous fleet of Plug-in Hybrid Electric Vehicles (PHEVs), which operate on two alternative fuels: electricity and compressed natural gas, for the purpose of waste collection problem. Two types of stations for charging and refueling stations are considered in the route service of the PHEVs.

Overall, the specific problems addressed in the WCVRP scenario with a single compartment are rather diverse. The main differences between these studies and the present work lie in the specific concepts of the waste collection problem. These differences include the type of vehicles used (e.g., compacting and visiting landfills), the availability of water refilling stations, multiple compartments, and the requirement for washing the bins.

The MC-WCVRP is an application of the well-known multi-compartment capacitated arc routing problem (MC-CARP) ([Küilerich and Wöhlk, 2018](#)). As different countries and cities handle waste collection differently, the original MC-CARP has been extended to several more complicated variants that seek to model the practice of waste handling. Most MCWCVRP studies consider a fixed compartment size associated with each waste type. This problem has not received much attention from researchers until now. Other applications include fresh food e-commerce ([Chen et al., 2020](#)), maritime transportation (e.g., [Christiansen et al., 2017](#); [Wang and Li, 2018](#); [Siswanto et al., 2019](#)), agricultural transportation (e.g., [Lahyani et al., 2015](#); [Sethanan and Pitakaso, 2016](#); [Tasar et al., 2019](#)), fuel distribution (e.g., [Urli and Kilby, 2017](#); [Kaabachi et al., 2019](#); [Hsu et al., 2020](#); [Yahyaoui et al., 2020](#)) and waste collection (e.g., [Rabbani et al., 2017](#); [Küilerich and Wöhlk, 2018](#); [Zbib and Laporte, 2020](#)). For a comprehensive review of MC-VRPs and their variants, we refer the reader to a recent survey paper by [Ostermeier et al. \(2021\)](#). The MC-WCVRP can be classified into two main categories, to be discussed as follows.

The first category considers a single planning day, which is the most studied case in the literature (e.g., [Muyldermans and Pang, 2010](#); [Reed et al., 2014](#); [Oliveira et al., 2015](#); [Rabbani et al., 2016](#); [Farrokhi-Aslet al., 2017](#); [Gajpal et al., 2017](#); [Zbib and Laporte, 2020](#)). The second category refers to the planning of multiple days. However, only a few studies have considered such a setting (e.g., [Elbek and Wøhlk, 2016](#); [Kiilerich and Wøhlk, 2018](#)). All MC-WCVRP studies considered a fleet of vehicles to collect only different types of waste. [Muyldermans and Pang \(2010\)](#) introduced the MCWCVRP and proposed a guided local search method to minimize the total variable costs. The results show the benefits of using multi-compartment vehicles, compared to single-compartment vehicles. [Reed et al. \(2014\)](#) developed an Ant Colony Optimization (ACO) algorithm to solve the Capacitated Vehicle Routing Problem (CVRP) utilizing a fleet of vehicles equipped with a fixed compartment size for traditional collecting recycling waste proposed by [Muyldermans and Pang \(2010\)](#). Later, [Oliveira et al. \(2015\)](#) studied a similar problem as in [Reed et al. \(2014\)](#) and developed a heuristic based on a clustering approach to solving a real-life case study of Portugal. A special type of recycling waste, namely the collection of different glass wastes, is studied by [Henke et al. \(2015\)](#) and [Henke et al. \(2019\)](#), which consider flexible compartment sizes. On the other hand, our research considers a scenario where a fixed compartment size is allowed. [Rabbani et al. \(2016\)](#) considered a mixed fleet composed of internal and external rented vehicles. A Genetic Algorithm (GA) was developed to minimize the total external rental vehicles and the total traveling and services costs. Later, [Rabbani et al. \(2017\)](#) extended the same problem to define the location of the treatment facilities and depot locations by using a Non-dominated Sorting Genetic Algorithm (NSGA-II). [Farrokhi-Asl et al. \(2017\)](#) considered a heterogeneous internal vehicles fleet to collect waste. Two multi-objective evolutionary metaheuristics, namely NSGA-II and a Multi-Objective Particle Swarm Optimization (MOPSO), were developed to solve this problem. Their paper considers a set of multiple depots and a set of waste disposal facilities that are waste compatible, and a vehicle visits the disposal sites only one time (only before returning to the depot), while in our research, only one depot and separation site are used to handle different waste.

[Zbib and Laporte \(2020\)](#) considered a fleet of compressed waste vehicles in which each compartment is associated with a compression factor for each waste type. An interesting study by [Yousra and Ahmed \(2022\)](#) considered MC-WCVRP with real-time information about the weights of the wastes provided, formulated the problem as a mixed-integer linear program, and proposed a heuristic to solve the problem. However, no experiment was conducted to demonstrate the effectiveness of their proposed method.

Other contributions to the literature are made by integrating Alternative Fuel Vehicles (AFVs) in the MC-WCVRP. These vehicles are common in practice due to their ability to reduce CO₂ emissions. [Gajpal et al. \(2017\)](#) used a fleet of AFVs with limited fuel tanks. The routes take into consideration the recharging fuel stations during their working services. Recently, [Erdem \(2022\)](#) considered heterogeneous traditional Electric Vehicles (EVs) (without compactor) with multi-compartments to

collect different waste and incorporated recharging stations in their solution framework. A HVNS was developed for the problem.

Concerning the multiple planning days, [Elbek and Wøhlk \(2016\)](#) considered glass and paper waste to be collected separately in each compartment of the vehicle. A special characteristic of their problem was that when the glass compartment was full, it could be swapped with another empty one. Also, they considered that a certain amount of waste in the landfill facility should be transported to the appropriate recycling facility. [Kiilerich and Wøhlk \(2018\)](#) investigated a Multi-Compartment Capacitated Arc Routing Problem (MC-CARP) by considering splitting and no-splitting waste demands, multi-day planning, and semi-periodic concepts. [Table 1](#) summarizes the main contributions of the studies to the MCWCVRP literature.

In the literature, the compaction operations and traveling activities to the separation facilities are typically studied separately. To our knowledge, there are only two papers ([Oliveira et al., 2015](#); [Zbib and Laporte, 2020](#)) considering compaction operations in MC-WCVRP, which were considered a key activity is waste collection ([Zbib and Laporte, 2020](#)). However, in [Zbib and Laporte \(2020\)](#), they considered that vehicles could only compact the waste without considering visiting separation/recycling facilities during the services for unloading the waste. In addition, their studies did not consider the maximum route duration, compaction time, and time windows. Unlike their study, we consider both the compaction of waste and the visits to disposal facilities during the services, providing further flexibility. What makes visiting separation sites during the vehicle service interesting and challenging is the trade-off between visiting these facilities and compacting wastes under the imposed maximum route duration and time windows constraints. Only a few articles consider compatible separation facilities ([Elbek and Wøhlk, 2016](#); [Rabbani et al., 2016](#); [Rabbani et al., 2017](#)). Under this setting, vehicles should visit all disposal facilities and empty each type of waste in its compatible separation facility. In other studies, [Henke et al. \(2015\)](#) and [Henke et al. \(2019\)](#), the waste is emptied only one time when the level of waste reaches the vehicle capacity. Our research aims to fill two gaps in the MC-WCVRP literature related to separation facilities and waste disposal. We consider scenarios where vehicles can empty different types of waste not only at any separation facility but also multiple times during their service. Besides, visiting refueling and recharging stations were considered by [Gajpal et al. \(2017\)](#) and [Erdem \(2022\)](#), whose studies allowed recharging vehicles to prevent the possibility of running out of fuel while it is in use. We adopt the problem setting of the aforementioned papers, which used two station types that can be visited multiple times, namely the separation facilities to empty the waste and recharge the EV as studied by [Erdem \(2022\)](#) or the AFV of [Gajpal et al. \(2017\)](#). This setting is similar to our current research (disposal sites set and water refilling stations to refill the water tank of the CRVBW). However, we note that multiple water stations are considered in our problem instead of using the depot to recharge AFVs as studied by [Gajpal et al. \(2017\)](#).

The aforementioned MC-WCVRP papers assumed bin wastes are only emptied from the collection bins. Compared to these papers, we allow that a subset of bins should be washed in addition to emptying all bin wastes at all collection points.

Our work incorporates several practical features, including a fleet of CRVBWs, water refilling stations, a maximum threshold value for un-washing bins, and the possibility to compact the wastes and/or visit the separation sites to empty the wastes into the traditional MC-WCVRP. Such constraints lead to problems with a high degree of relevance to reality. Thus, our problem adds new features that increase the computational complexity compared to those studied in the existing studies.

Table 1

Key characteristics of MC-WCVRPs in the literature, classified according to [Ostermeier et al. \(2021\)](#)

Reference	MD/SD	OB	DC	NV	H/HT	Fix	Flex	MRF	CF	TH	WRS	BW
Muyldermans and Pang (2010)	SD	Min. total variable total costs	√	CV	HT	√	-	-	-	-	-	-
Rabbani et al. (2016)	SD	Min. total costs	√	CV	HT	√	-	√*	-	-	-	-
Reed et al. (2014)	SD	Min. total traveled distance	-	CV	H	√	-	-	-	-	-	-
Oliveira et al. (2015)	SD	Min. total traveled distance	-	CV	H	√	-	-	√	-	-	-
Henke et al. (2015)	SD	Min. total costs	-	CV	H	-	√	-	-	-	-	-
Henke et al. (2019)	SD	Min. total traveled distance	-	CV	H	-	√	-	-	-	-	-
Rabbani et al. (2017)	SD	Min. total costs	√	CV	HT	√	-	√*	-	-	-	-
Farrokhi-Asl et al. (2017)	SD	Min. total costs	√	CV	HT	√	-	√*	-	-	-	-
Gajpal et al. (2017)	SD	Min. total traveled distance	√	AFV	H	√	-	-	-	-	-	-
Zbib and Laporte (2020)	SD	Min. variable transportation costs	-	CV ⁴	HT	√	-	-	√	-	-	-
Erdem (2022)	SD	Min. total travel costs	√	EV	HT	√	-	-	-	-	-	-
Elbek and Wöhlk (2016)	MD	Min. total costs	-	CV	H	√	-	√*	-	-	-	-
Kiilerich and Wöhlk (2018)	MD	Min. total costs	-	CV	H	√	-	-	-	-	-	-
This research	SD	Min total costs	√	CRVBW	H	√	-	√	√	√	√	√

MD: Multiple days; SD: Single day; OB: Objective function; DC: Maximum route duration/distance; **NV:** Vehicle type (Conventional Vehicle (CV), or Alternative Fuel Vehicle (AFV)); **H:** Homogeneous vehicles; **HT:** Heterogeneous vehicles; **Fix:** Fixed size compartments; **Flex:** Flexible size compartments; **CF:** Compaction waste Factor; **MRF:** Multiple Recycling Facilities is considered during the services in the studies or not; **TH:** Threshold concept; **WRS:** Water Refilling Stations; **BW:** Bin Washer is considered or not

* The recycling facilities are visited only before returning to the depot

Given that drivers' lunch breaks are typically considered idle times, MC-WCP-BW is relevant to VRPs with breaks and rest periods. Alternatively, incorporating a break at a node into transportation planning reduces driving time, which is similar to incorporating a washing and/or compaction break at a node, as considered in our study.

Several studies have considered break/rest periods in the context of the dial a ride problem ([Zhang et al., 2015](#); [Masmoudi et al., 2016](#) ; [Lim et al., 2017](#)), VRP with time windows using breaks and shifts ([Karademir et al., 2020](#)), home health care routing problems ([Trautsamwieser et al., 2011](#); [Fikar and Hirsch, 2015](#); [Liu et., 2020](#); [Haitam et al., 2021](#)), vehicle routing problem with lunch break ([Buhrkal etl., 2012](#) ; [Coelho et al., 2016](#)), electric vehicle routing problem ([Cortés-Murcia et al., 2023](#)), and workforce scheduling and routing problem ([Kovacs et al., 2012](#) ; [Liu et al., 2017](#) ; [Villegas et al., 2018](#)).

3. Problem Definition

The problem MC-WCP-BW involves a graph $G=(V,A)$ where V represents the set of vertices consisting of a starting depot 0, a set of collection points N , a set of separation sites S , a set of water refilling stations W , and an ending depot $n' = n + 1$ (the location of vertex n' coincides with the location 0), where $n=|N|$. We have $V = \{0\} \cup \{n'\} \cup N \cup S \cup W$. A travel cost c_{ij} , a travel distance d_{ij} and a travel time t_{ij} are associated with each arc $(i,j) \in A$.

Let $P = \{1, \dots, n_p\}$ be a set of waste types. A set $\{q_i^1, \dots, q_i^{n_p}\}$ of waste volume is associated with collection point i , each quantity corresponding to a specific type of waste. In other words, each collection point i is associated with the same number of bins.

Each vertex $i \in V \setminus \{0, n'\}$ is also associated with the following parameters:

- If $i \in N$, a service time S_i^N represents a fixed time associated with visiting collection point $i \in N$.
- If $i \in S$, a service time S_i^S represents a fixed time associated with separation site i .
- If $i \in W$, a service time S_i^W represents a fixed time associated with water refilling stations i .

The separation site $s \in S$ can treat all waste types.

An available fleet of n_v homogeneous CRVBW is based at the start depot 0 to serve the collection points' requests. Each vehicle has a capacity Q of the water tank and n_p compartments, each one of which is associated with a specific type of waste $p \in P$ and can contain only waste of this type, up to a maximum capacity of C^p . We note that all CRVBW have an equal number of compartments.

For each collection point $i \in N$, a flag parameter θ_i is given in input. If θ_i is equal to 1, all the bins located at i must be washed, whereas if $\theta_i = 0$, washing is not mandatory; however, if washing takes place, then all bins located at $i \in N$ must be washed. Each collection point's washing operation consumes a fixed quantity of water ζ . Moreover, to provide a specific service level to the company, we set a maximum number of collections that can remain unwashed ψ . A fixed cost F^r is incurred each time the vehicle refills its tank from the water refilling stations.

A vehicle route is a path in graph G starting at vertex 0 and ending at the same depot $n' = n + 1$, visiting a set $\bar{N} \subseteq N$ of collection points and such that:

- Each time a collection point i in \bar{N} is visited, the compartments are loaded with the waste from the corresponding bin. In addition, we assume that all its bins are emptied. This implies that a collection point must be visited only once.
- The total duration of the route computed as the sum of the traveling and service times is not greater than a given limit T_{max} .
- Compartment capacity can never be violated during the loading operation.
- At each washing of a bin at a collection point $i \in N$, a fixed time S_i^B is considered.
- The vehicle can refill its water tank at the water refilling station.
- To perform washing, the vehicle must have a sufficient remaining volume of water.

- At each visiting water refilling station, the tank is refilled at its maximum.
- Vehicle compartments can be emptied at separation sites.
- Each separation site can receive all different types of waste.
- On its route, the vehicle can perform compact operations for the different compartments. A compaction operation of the compartment r with a load equal to x reduces x to $x\vartheta^r$, where $0 < \vartheta^r < 1$ is the compact ratio. A compaction operation requires S^c time units. In addition, a fixed cost F^c is associated with each compaction operation. This cost represents the fuel cost consumed at the compaction operation. We assume that a vehicle can perform at most one compacting operation before visiting a separation site. More specifically, two consecutive compaction operations of the same waste type $p \in P$ are not allowed. The vehicle must visit a separation site to empty the waste type p if it is already compacted, then the compaction of the same waste type p is allowed.
- The vehicle leaves and returns to the same depot with an empty load.
- The vehicle must visit the separation sites that correspond to all their compartments at least once per route.

In this problem, the separation and water refilling stations may be visited several times during each route. Using traditional formulation of the Green Vehicle Routing Problem (GVRP) and the Electric Vehicle Routing Problem (EVRP), a set of dummy vertices to permit multiple visits at each vertex in the set of separation sites S and the set of water refilling stations W are required. However, additional dummy vertices can increase the number of variables in the MILP and consume longer computational times (Bruglieri et al., 2019); especially in this work, two dummy sets vertices are considered, contrary to a single one that is widely applied in the GVRP and EVRP (e.g., Erdoğan and Miller-Hooks, 2012; Schneider et al., 2014; Hiermann et al., 2016; 2019; Pelletier et al., 2019; Nolz et al., 2022). Thus, our problem is considered more challenging. One of the main contributions of our research is to develop a cloneless formulation based on a multigraph in which separation and water refilling stations are not explicitly modelled. This formulation is based on the GVRP reformulation of Bruglieri et al. (2019). However, two main differences are considered. First, we consider several possible paths utilize an arc., In other words, we allow visiting one or more than one different station whereas in Bruglieri et al. (2019) only one station is allowed, where our problem is more challenging and complex but provides further routing flexibility. Second, in Bruglieri et al. (2019), the consumption of energy depends on the distance traveled over an arc. However, in our case, the refilling is dependent only on the node.

The nodes in the graph consist of a set of collection points, an initial depot, and an end depot with several paths starting from node i visiting a subset of separation sites S and/or a subset of water refilling stations W .

Let a refill path from $i \in V$ to $j \in V$ and a separation path from $i \in V$ to $j \in V$, are simple paths in G , starting from node i visiting a subset of separation sites S and a subset of water refilling stations

W and ending at node j . For each collection points pairs (i, j) with $i \neq j$, we denote by H_{ij} and F_{ij} the index sets of all refill and separation paths from the two collection points i and j . For refill path H and separation path F , the total travel $d(H)$ and $d(F)$ are equal to the sum of the total distance of the arcs it traverses. We redefine A the arc set containing arcs form (i, h, f, j) , where $i, j \in N, \forall h \in H_{ij}$ and $\forall f \in F_{ij}$. Each arc (i, j) forming A can be represented as a possible sequence of consecutive visits format $k \in K = \{1, 2, 3, 4\}$.

The following arc types are as follows:

- $k=1$; if there is a direct travel between two collection points $(i, 0, 0, j)$
- $k=2$; if there is a proper refilling path $(i, h, 0, j)$ that visits only a water refilling station
- $k=3$; if there is a proper separation path $(i, 0, f, j)$ that visits only a separation site
- and $k=4$; if there is a refilling and separation path (i, h, f, j) that visits both a water refilling station and a separation site. This type is constructed by selecting the pair of water refilling station and separation facilities (h, f) , implying the minimum detour between i and j . We note that there is no restriction in the order in which separation site and water refilling station must be visited.

Figure 2 shows the different possibilities for traveling of the vehicle through two collection points i and j .

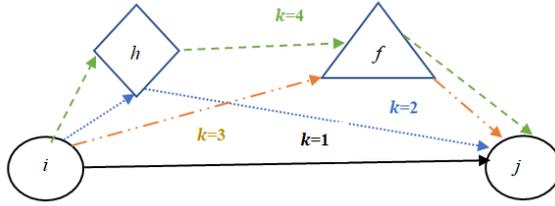


Fig. 2: Example of the different possible paths between two collection points

Each arc $a=(i, h, f, j) \in A$ is associated with a travel distance $d(a)=d(i, h, f, j)$ and a travel time $t(a)=t(i, h, f, j)$ and a routing cost $c(a)=d(i, h, f, j)$. We define d_{ij}^k and t_{ij}^k , the extra traveling distance and time, respectively to the water refilling station and/or separation site for link options $k=2, 3$ and 4 along link $i - j$. Let $d_{ij}^2 = \min\{d_{iw}+d_{wj}-d_{ij}\}$ representing the minimum detour to the water refilling station $w \in W$ between two collection nodes i and j . We suppose that $d_{ij}^2 = t_{ij}^2$. Similarly, to the refilling stations, we define $d_{ij}^3 = \min\{d_{is} + d_{sj} - d_{ij}\}$ representing the minimum detour to the separation site $s \in S$ between two collection nodes i and j . We assume that $d_{ij}^3 = t_{ij}^3$. Finally, we define $d_{ij}^4 = \min\{d_{iw}+d_{ws} + d_{sj}-d_{ij}; d_{is}+d_{sw} + d_{wj}-d_{ij}\}$ that represents the minimum detour to the water refilling station $w \in W$ and separation site $s \in S$ between two collection nodes i and j . Similar to the previous arc types $k=2$ and 3 , we assume that $d_{ij}^4 = t_{ij}^4$.

The final graph does not contain the sets of water refilling stations and separation sites. The main idea is to define four types of an arcs (i, j) , $i, j \in N \cup \{0\} \cup \{n'\}$, where a vehicle directly travels from i to j or have stops at the water refilling stations and/or separation sites in between. Depending on whether between i and j a separation site and/or a water refilling station is visited, the capacity of the vehicle compartments and the water tank will be reset. For every pair of i, j , the minimal detour is identified, and the additional travel time and distance are calculated. As a result, the travel time and distance from i to j can have different values based on the detour case. Depending on the decision if a visit to a separation and/or water refilling station is planed between i and j , the problem can be modelled on a graph without the vertices of separation sites and water refilling stations by introducing two additional binary decision variables indicating the visits. We define by δ_j^p that indicates whether compacting operations already took place before node j or not and help avoid multiple compacting operations on already compacted waste. And we define λ_j^p is tracking if the compacting operations take place at node j and is used to compute the residual volume of the compacted waste which clearly depend on the amount of waste carried in the vehicle when reaching j .

The cost of a route is equal to the sum of the traveling costs of the arc set traversed by the route, the fixed costs associated to refill the vehicle at the water refilling station, and the fixed costs of the compact operations. The MC-WCP-BW consists of designing a set of vehicle routes of the minimum total cost, including the travel distance, compaction, and refilling costs, such that each collection point is visited exactly once by exactly one route. All notations and used variables are presented in [Table 2](#).

Table 2
Definitions of parameters and variables in the MC-WCP-BW model

$0, n'$	Start and end vertices of a vehicle route
N	Set of collection points
S	Set of separation sites
W	Set of water refilling stations
K	$K = \{1, 2, 3, 4\}$: direct ($k = 1$), refill ($k = 2$), separation ($k = 3$) and refill and separation ($k = 4$) link options
V	Set of nodes where $V = \{0\} \cup \{n'\} \cup N \cup S \cup W$
P	Set of waste types
c_{ij}^k	Cost between nodes i and j using path type $k \in K$ where i and $j \in N \cup \{0\} \cup \{n'\}$
t_{ij}	Travel time between nodes i and j where i and $j \in N \cup \{0\} \cup \{n'\}$
d_{ij}	Distance between nodes i and j where i and $j \in N \cup \{0\} \cup \{n'\}$
t_{ij}^2	Minimum detour time between two collection point nodes i and j passing through a water refilling station, where i and $j \in N$

t_{ij}^3	Minimum detour time between two collection point nodes i and j passing through a separation site, where i and $j \in N$
t_{ij}^4	Minimum detour time for both the water refilling station and separation site between two collection points i and j where $i, j \in N$
a_{ij}^2	Minimum detour distance between two collection point nodes i and j passing through a water refilling station, where i and $j \in N$
a_{ij}^3	Minimum detour distance between two collection point nodes i and j passing through a separation site, where i and $j \in N$
a_{ij}^4	Minimum detour distance for both the water refilling station and separation site between two collection points i and j where $i, j \in N$
Q	Tank capacity
S_i^N	Fixed service time necessary to empty the waste bin at collection point $i \in N$
S_i^S	Fixed service time necessary to empty the wastes at the separation site node $i \in S$
S_i^W	Fixed service time necessary to refill the vehicle at the water refilling station $i \in W$
S_i^B	Fixed service time necessary to wash a bin at node $i \in N$
S^c	Fixed amount of time necessary for the compaction operation
C^p	Compartment capacity of the waste type $p \in P$
ζ	Water consumption rate to wash the bins of a collection point
ψ	Threshold for unwashed bins
F^r	Fixed refilling water cost at the water refilling station
F^c	Fixed compaction cost
q_i^p	Quantity of waste type p at the collection point node i
ϑ^p	Compaction rate for waste of type $p \in P$
n_v	Available number of vehicles
T_{max}	Maximum duration in time of a vehicle route
θ_i	Flag parameter is equal to 1 if and only if bins located at collection point $i \in N$ must be washed and 0 otherwise
x_{ij}^k	Binary variable is equal to 1 if the vehicle travels from $i \in N \cup \{0\}$ to $j \in N \cup \{n\}$ following a sequence visit $k = \{1, 2, 3, 4\}$, and 0 otherwise
δ_i^p	Binary variable that is equal to 1 if and only if the vehicle has already compacted the waste of type $p \in P$ before reaching node $i \in N$, and 0 otherwise
λ_i^p	Binary variable that is equal to 1 if and only if a compaction operation is executed at node $i \in N$ regarding waste type $p \in P$, and 0 otherwise
b_i	Binary variable that is equal to 1 if and only if a bin is washed at collection point $i \in N$, and 0 otherwise
z_{ij}	Binary variable that is equal to 1 if a refilling operation takes place between $i \in N$ and $j \in N$, and 0 otherwise

l_{ij}	Binary variable is equal to 1 if and only if a separation site is visited between $i \in N$ and $j \in N$, and 0 otherwise
τ_i	Nonnegative continuous variable that represents the arrival time at node $i \in N \cup \{0\} \cup \{n'\}$
o_j	Nonnegative continuous variable represents the tank water level when visiting node $j \in N$
u_i^p	Nonnegative continuous variable that represents the load of compartment $p \in P$ immediately after servicing $i \in N$

We provide the following mixed-integer programming formulation for the MC-WCP-BW:

$$\text{minimize } \sum_{k \in K} \sum_{i \in N \cup \{0\} \cup \{n'\}} \sum_{j \in N \cup \{0\} \cup \{n'\}} c_{ij}^k x_{ij}^k + \sum_{i \in N \cup \{0\} \cup \{n'\}} \sum_{j \in N \cup \{0\} \cup \{n'\}} F^r z_{ij} + \sum_{p \in P} \sum_{i \in N} F^c \lambda_i^p \quad (1)$$

subject to

$$\sum_{k \in K} \sum_{j \in N \cup \{n'\}} x_{ij}^k = 1 \quad \forall i \in N \quad (2)$$

$$\sum_{k \in K} \sum_{j \in N \cup \{0\}} x_{ji}^k = 1 \quad \forall i \in N \quad (3)$$

$$\sum_{k \in K} x_{0i}^k \leq 1 \quad \forall i \in N \quad (4)$$

$$\sum_{k \in K} x_{in'}^k \leq 1 \quad \forall i \in N \quad (5)$$

$$\sum_{k \in K} \sum_{i \in N} x_{0i}^k \leq n_v \quad (6)$$

$$z_{ij} = x_{ij}^2 + x_{ij}^4 \quad \forall i, j \in N \cup \{0\} \cup \{n'\} \quad (7)$$

$$l_{ij} = x_{ij}^3 + x_{ij}^4 \quad \forall i, j \in N \cup \{0\} \cup \{n'\} \quad (8)$$

$$\tau_j \geq \tau_i + (S_i^N + t_{ij}) \sum_{k \in K} x_{ij}^k + \sum_{p \in P} S^C \lambda_i^p + S_i^S l_{ij} + S_i^W z_{ij} + S_i^B b_i + t_{ij}^2 z_{ij} + t_{ij}^3 l_{ij} + t_{ij}^4 z_{ij} + t_{ij}^4 l_{ij} - M(1 - \sum_{k \in K} x_{ij}^k) \quad \forall j \in N \cup \{n'\}, \forall i \in N \cup \{0\} \quad (9)$$

$$\tau_0 = 0 \quad (10)$$

$$\tau_j \leq T_{\max} - (s_j^N + t_{jn'}) - (t_{jn'}^3 + S_s^S) \quad \forall j \in N, s \in S \quad (11)$$

$$u_i^p \geq q_i^p (1 - \lambda_i^p) + (v^p q_i^p) \lambda_i^p \quad \forall i \in N, \forall p \in P \quad (12)$$

$$u_j^p \geq u_i^p + q_j^p - M(1 - \sum_{k \in \{1,2\}} x_{ij}^k) - M \lambda_j^p \quad \forall i, j \in N, \forall p \in P \quad (13)$$

$$u_j^p \geq (u_i^p + q_j^p) v^p - M(1 - \sum_{k \in \{1,2\}} x_{ij}^k) - M(1 - \lambda_j^p) \quad \forall i, j \in N, \forall p \in P \quad (14)$$

$$\sum_{p \in P} (u_0^p + u_{n'}^p) = 0 \quad (15)$$

$$u_i^p \leq C^p \quad \forall i \in N, \forall p \in P \quad (16)$$

$$o_j \leq o_i - \eta b_i + Q z_{ij} + M(1 - \sum_{k \in K} x_{ij}^k) \quad \forall i \in N \cup \{0\}, j \in N \cup \{n'\} \quad (17)$$

$$o_i \leq Q \quad \forall i \in N \quad (18)$$

$$o_0 = Q \quad (19)$$

$$\delta_0^p = 0 \quad \forall p \in P \quad (20)$$

$$\delta_j^p \leq M(1 - l_{ij}) \quad \forall i \in N \cup \{0\}, \forall j \in N \cup \{n'\}, \forall p \in P \quad (21)$$

$$\delta_j^p \geq \delta_i^p - M(1 - \sum_{k \in \{1,2\}} x_{ij}^k) - M l_{ij} \quad \forall i \in N \cup \{0\}, \forall j \in N \cup \{n'\}, \forall p \in P \quad (22)$$

$$\delta_j^p \geq \lambda_i^p - M(1 - \sum_{k \in \{1,2\}} x_{ij}^k) - M l_{ij} \quad \forall i \in N \cup \{0\}, \forall j \in N \cup \{n'\}, \forall p \in P \quad (23)$$

$$\lambda_i^p \leq 1 - \delta_i^p \quad \forall i \in N, \forall p \in P \quad (24)$$

$$b_i \geq \theta_i \quad \forall i \in N \quad (25)$$

$$\sum_{i \in N} b_j \geq n - \psi \quad (26)$$

$$\sum_{k \in \{3,4\}} x_{ij}^k \leq 1 - \lambda_i^p \quad \forall i, j \in N, \forall p \in P \quad (27)$$

$$\sum_{p \in P} (\lambda_0^p + \lambda_{n'}^p) = 0 \quad (28)$$

$$\sum_{i \in N} (l_{0i} + z_{0i}) = 0 \quad \forall i \in N \quad (29)$$

$$x_{ij}^k \in \{0, 1\}; z_{ij} \in \{0, 1\}; l_{ij} \in \{0, 1\} \quad \forall i, j \in N, \forall k \in K \quad (30)$$

$$b_i \in \{0, 1\} \quad \forall i \in N \quad (31)$$

$$\lambda_i^p \in \{0, 1\}; \eta_i^p \in \{0, 1\}; \delta_i^p \in \{0, 1\}; \tau_i, u_i^p \geq 0 \quad \forall i \in N, \forall p \in P. \quad (32)$$

In the above model, M is a large constant. The objective function (1) consists of two terms to be minimized. The first term represents the total costs of traveling and refilling, while the second corresponds to the total compaction cost. Constraints (2) and (3) impose that each collection point is visited by exactly one vehicle with a corresponding sequence visit option. Constraints (4) and (5) states that at most one sequence visit option k is selected for the links emanating depot 0 and the links entering depot n' , respectively. Constraint (6) guarantees that the maximum number of allowed vehicles is respected. Constraints (7) ensure that the refilling water operation. In other words, this constraint enforces the water refilling operation between a pair of collection points for only two cases of a possible sequence of visits ($k=2$ and $k=4$). Constraints (8) enforce the visiting of the separation site to empty the wastes between two collection points i and j only for the path type $k=3$ and $k=4$. Constraints (9)-(11) evaluate the arrival times at the different vertices and impose the maximum duration T_{max} of the vehicle route by ensuring its visit to a separation site. Constraints (12)-(16) impose the vehicle capacity constraints. Specifically, constraints (12) impose that the load of compartment p immediately after servicing i depends on the quantity q_i^p and the execution ($\lambda_i^p=1$) or not ($\lambda_i^p=0$) of a compaction operation. **Indeed, if $\lambda_i^p=1$ the corresponding constraint reads $u_i^p \geq q_i^p$, otherwise if $\lambda_i^p=0$ it reads $u_i^p \geq \vartheta^p q_i^p$.** Constraints (13) and (14) are active only if link $i-j$ is used with $k \in \{1, 2\}$ (i.e., $\sum_{k \in \{1, 2\}} x_{ij}^k = 1$). If they are not active, constraints (12) are imposed, also in the case which $\sum_{k \in \{3, 4\}} x_{ij}^k = 1$. If a compaction operation is executed at j ($\lambda_j^p=1$), constraints (13) are redundant whereas constraints (14) impose:

$$u_j^p \geq (u_i^p + q_j^p) \vartheta^p, \quad (33)$$

i.e., the vehicle leaving j has compacted a total waste computed as the sum of the waste arriving at i plus the demand at q_j^p . **Viceversa, if $\lambda_i^p=0$, constraints (14) are redundant whereas constraints (13) impose:**

$$u_j^p \geq u_i^p + q_j^p. \quad (34)$$

Constraints (15) state that a vehicle leaves the start depot 0 and enters the end depot n' with a total load equal to 0. Constraints (17)-(19) impose the vehicle- water tank capacity constraints. Constraints (20)-(24) are responsible for determining the values of the variables δ . Constraints (21) specify that $\delta_j^p \leq 0$, i.e., $\delta_j^p = 0$, whenever arc (i, j) is used in the solution and $l_{ij}=1$, indicating a

separation along the link $i - j$. Otherwise, these constraints are redundant. Constraints (22) and (23) state that $\delta_j^p \geq 0$ if arc (i, j) is not used or if $l_{ij}=1$. However, if the arc is used and $l_{ij}=0$, these constraints enforce $\delta_j^p \geq \min \{\delta_i^p, \lambda_i^p\}$. More specifically, Constraints (22) state that if while traveling from i to j the vehicle does not visit a separation site, then the compaction variable $\delta_j^p \geq \delta_i^p$. We can restore the compaction option only by visiting a separation site. The constraint has no binding effect if the vehicle is not moving from i to j . Constraints (23) state that if the vehicle has compacted the waste in i and the vehicle traveling between i and j , then when the vehicle arrives in j the vehicle has already compacted the waste and therefore the compaction option is not available anymore. The constraint has no binding effect if the vehicle is not moving from i to j . Finally, constraints (24) dictate that compaction at node i for waste p can only occur if $\delta_i^p = 0$, meaning that the waste has not been compacted before reaching node i . In other words, Constraints (24) state that at each node, the vehicle cannot compact the waste if the compaction option is not available, or in other words, if the vehicle has already compacted the waste along the route and did not visit a separation site after the last compacting operation.

Constraints (25) and (26) impose the constraints on the number of bins to be washed and the bins that must be washed, respectively. Constraints (27) impose that if a compaction operation is executed at i , no separation can be executed along any arc emanating from i , whereas constraints (28) impose that no compact operation are executed at the start and end depots. Constraints (29) impose that no compaction and separation operations are executed along links from depot 0 and the collection points. Finally, the domains of the variables are defined as from the list of decision variables in constraints (30) - (32).

4. Hybrid Adaptive Variable Neighborhood Search for the MC-WCP-BW

The MC-WCP-BW is a new and challenging problem. The MC-WCVRPTW is also an NP-hard problem (Muyldermans and Pang, 2010); the MC-WCP-BW is a generalization of the WCVRPTW, which is also NP-hard problem. Thus, it is difficult to solve to optimality in large-size instances using exact solution methods. The majority of the studies have been solved using different metaheuristics approaches (see, e.g., Reed et al., 2014; Rabbani et al., 2016; Farrokhi-Asl et al., 2017; Henke et al., 2019; Kaabachi et al., 2019; Zbib and Laporte, 2020; Yahyaoui et al., 2020; Chen et al., 2020; Erdem, 2022). Therefore, we propose a HVNS algorithm to solve the MC-WCP-BW.

While several established (H)VNS algorithms have been developed, our HVNS method differs from the approach proposed by Chen et al. (2020) in several ways. Firstly, our initial solution does not include a compaction operation or visits to separation sites and water refilling stations. To insert unserved customers (collection points), we identify the best position in the partially constructed route that uses an available vehicle while respecting the model constraints. In contrast, Chen et al. (2020) used a scanning process to evaluate each unserved customer against the compartment capacity limit and the time window limit and applied a greedy algorithm to insert the customer into a route if the constraints are met.

Secondly, we use an intelligent adaptive mechanism to re-order the neighborhood search structure at each shake, whereas [Chen et al. \(2020\)](#) demonstrated two path relinking approaches. The first approach reconnects the current solution with the personal best solution, while the second reconnects the current solution with the global best solution. Thirdly, we apply four local search operators that use a re-ordering approach (similar to the shaking phase) at each iteration, whereas [Chen et al. \(2020\)](#) applied local search operators with the same order at each iteration and retain the optimal operator solution of these operators. Fourthly, when the newly created solution after the shaking and improvement phase is not better than the existing best solution, our HVNS constructs a new solution by employing the crossover operator between the existing best solution and a new solution created by the constructive heuristic, whereas [Chen et al. \(2020\)](#) generated a new solution based on the same current solution using neighborhood operators. Fifthly, in their method, a newly generated solution is accepted only if its objective value is higher than the present one, whereas, in our approach, a new solution is approved if it improves the objective value or meets the Cauchy criterion. Finally, instead of a predetermined number of iterations, our stopping criterion is a preset number of iterations without improvement. Our research's unique approach can guide route planners and managers to optimize their waste transportation activities using innovative waste collection vehicles.

VNS can be divided into two phases: (i) a deterministic phase which a local search makes it possible to converge towards a local optimum, and (ii) a stochastic (shaking) phase to escape from a local optimum. The shaking phase generates a new solution x' from the current solution x according to a given neighborhood. The traditional VNS uses an initial solution x as the starting point, and a set of $N_{l_{max}}$ neighborhoods $N = \{N_1, \dots, N_{l_{max}}\}$. At each iteration, a solution random x' is generated using the current neighborhood N_l . Then, a local search is applied on x' , which generates a new solution x'' . If this new solution x'' is better than x' , an update is performed, and the process resumes with the first neighborhood. Otherwise, the same steps are repeated for the next neighborhood N_{l+1} .

- Our approach is different from other hybrid VNS approaches in the literature. We incorporate several advantages of well-established metaheuristic methods, including the GA crossover operator and a flexible temperature parameter control inspired by SA. During diversification and intensification, we use intelligent adaptive mechanisms. Traditional hybridization strategies incorporate VNS into a single or population-based method or use a single diversification/intensification mechanism. However, our strategy aims to increase VNS performance and convergence to optimal solutions. We deliberately introduce these extra strategies to maximize our advantage. Our enhanced HVNS is not only adapted to MC-WCP-BW, but it can also be seen as a new general algorithmic framework.
- In the traditional VNS, a new solution is constructed from the current solution using a predefined neighborhood search with a fixed order (N_h). This solution can be accepted or rejected, and the same procedure is repeated. However, our HVNS enhances this mechanism with an adaptive and

intelligent approach. We start with the predefined order of all the neighborhood searches, as in traditional VNS. Then, the current neighborhood is considered for the current solution x . If the new solution x' has better quality than the current solution x , we change the rank order and adopt the current operator as the first in the new order. Otherwise, we move to the next operator. We use the same adaptive mechanism in the intensification phase, which applies to the local search operator. By using these two techniques, we can identify and explore more potential regions in the search space, leading to better solutions and more time for the most successful operators.

- In a typical VNS algorithm, the cooling factor for reducing the temperature at each iteration is based on the temperature function used in SA. However, in our HVNS approach, we adopt a more flexible temperature control strategy. Specifically, if the current best solution x_{best} has not been improved after a preset number of runs, we adjust the current temperature to a lower value. This approach is designed to address the problem of quick convergence to suboptimal local solutions or slow convergence to acceptable solutions that do not meet the desired level of the objective function.
- In most VNS algorithms, when a newly generated solution is not better than the current solution, the algorithm typically generates a new solution using neighborhood operators on the same current solution. However, in our approach, we also use the crossover operator, a fundamental property of genetic algorithms, to construct a new solution when necessary. Specifically, we perform the crossover operation between the current best solution (x_{best}) and a new solution generated by the constructive heuristic, which is known for its high diversity. This results in a starting solution for VNS restart that incorporates information from both the best solutions discovered by the algorithm and the constructive heuristic. This approach provides a balanced approach between intensification around the best solution and diversification to explore new regions of the search space in a controlled manner instead of arbitrary methods. Our technique has been successfully applied in various metaheuristics (e.g., [Masmoudi et al., 2021; 2022](#)).

The shaking HVNS component consists of four neighborhood structures and four operators in the improvement phase. The improvement phase is comprised of four classical operators that have been modified to improve the quality of solutions received from the shaking phase for each new generation. The structure is illustrated in [Algorithm 1](#). An initial solution developed using an insertion heuristic ([Section 4.1](#)) is used as input for the shaking and improvement phases. The searching and improvement process runs for a number of iterations until a stopping criterion is met. During the shaking phase, shakes are conducted on a solution x' , which is a replica of x , to find a better solution. Each shake involves a single move chosen from a neighborhood structure. There are four neighborhoods with structure movements ([Section 4.2](#)), where the first and second structures consist of swapping and inserting one or a pair of collection points in between routes, respectively. The latter two neighborhoods are comprised of swapping segments of successive collecting points. The size of these two structures is set at two to three successive collecting points. An intelligent adaptive mechanism is used during the shaking phase

to improve the discovery of the most advantageous neighborhood throughout each iteration. This system takes into account the present order and the maximum number of shaking operators. If an improvement is detected in the previous iteration, the shaking operators are reordered in decreasing order.

After obtaining the newly created solution, the improvement step is initiated. The improvement phase involves selecting a local search operator from four local search operators (Section 4.3) to quickly and efficiently enhance the solution acquired from the present neighborhood. The intelligent adaptive procedure used for the four local searches is likewise used in the improvement phase, as in the shaking phase. At the end of each application of a local search operator, redundant water refilling stations, separation site nodes, and compaction operations may occur. To keep the method feasible in terms of compaction waste and refilling/separation constraints (Section 4.4), four particular operators are used. If the newly obtained solution after applying the solution operations (i.e., neighborhood construction, local search, feasibility operators of compaction) and imposing the refilling/separation constraints is not better than the current solution, a new solution is constructed using a crossover operator that takes the current best solution found so far and a newly generated constructive heuristic as inputs.

Algorithm 1: Overview of the HVNS

```

1. Procedure HVNS (initial solution  $x$ , best solution  $x_{best}$ , neighborhood structures  $N_h$ , local search operators  $L$ ,
   feasibility operators, ShakingOrder; ImprovementOrder)
2. Repeat
3.    $OrderN \leftarrow ShakingOrder$ ; // initialize the order of the neighborhood structures
4.    $OrderI \leftarrow ImprovementOrder$ ; // initialize the order of the local search
5.   Repeat
6.      $x' = RandomNeighbor(x, h, ShakingOrder)$  // shaking phase
7.      $x'' = Local\_Search(x', l, ImprovementOrder)$  // improvement phase
8.     Apply the feasibility procedure // Apply the four operators related to the compaction
                                     operation, removing, insertion of the refilling/separation sites
9.     If  $f(x'') < f(x)$  Then
10.       $x \leftarrow x''$ ;
11.       $h = 1$ ;
12.      Apply intelligent re-ordering shaking phase; // update the order of the neighborhood searches
                                                       of the shaking phase
13.      Apply the intelligent re-ordering improvement phase; // update the order of the local searches
                                                             of the improvement phase
14.      If  $(x > x_{best})$  Then
15.         $x_{best} = x$ ;
16.      Else
17.        Construct a new solution; // construct a new solution using crossover based on  $x$  and  $x_{best}$ 
18.      End Else
19.    Else
20.       $h \leftarrow h + 1$ ;
21.    Until  $h \leftarrow h_{max} + 1$ ;
22. End procedure

```

The detailed steps of the HVNS are shown in Algorithm 2. Let the temperature T be set to its maximum value T_{max} , N_l be a set of neighborhoods $l = \{1, \dots, N_{l_{max}}\}$, N_h be a set of neighbourhood searches $h = \{1, \dots, h_{max}\}$, and x be the initial solution, and x_{best} be the current best solution, initialized to x . In addition, the algorithm receives an initial predefined order of neighbourhoods *ShakingOrder* and local searches *ImprovementOrder*.

If the proposed HVNS runs for a certain number of iterations without improvement, then the following operations are performed. First, an effective construction heuristic generates the initial solution (see Section 4.1). Then, the following steps are performed. A new solution x' is generated using the current neighborhood search N_h . At each iteration, the neighborhood operators are re-ordered in descending order, based on the improvements made when applying each of them at the previous iteration. An initial predefined neighborhood order (in the order of most complex to the least complex) is adopted. Then, for each diversification phase, the neighborhood operators (see Section 4.2) are successively adopted in their order (lines 15-19 of Algorithm 2), and the incumbent solution, x is diversified (line 10 of Algorithm 2) to construct a new solution x' . The order of neighborhood search is reset to its initial one at each HVNS run (line 3 of Algorithm 2). The new solution is enhanced by a powerful local search procedure (see Section 4.3) to construct the next solution x'' . The selection of the local search is pre-determined based on the performance score obtained at the next iteration (lines 20-24 of Algorithm 2), in a similar manner in most of the shaking phase. If the objective value offered by solution x'' is less than the one offered by solution x or accepted by the acceptance function, x'' is set to the new current solution. We remarked that the acceptance function (line 12 of Algorithm 2) is expressed by the Cauchy function of Tiwari et al. (2006): $T / (T^2 + \Delta E^2)$. This function is evaluated by randomly drawing β between 0 and 1 and replacing x by x' if $\beta < T / (T^2 + \Delta E^2)$, where ΔE is the difference in costs resulting from the current and the next solutions. The temperature is reduced only when x_{best} is not improved at the last iteration using the cooling scheme: $T = \alpha \times T$, where α is the cooling factor (line 6 of Algorithm 2).

If the objective value of the solution x'' (i.e., $f(x'')$) is smaller than the objective value of the current best solution x_{best} (i.e., $f(x_{best})$), the new current solution x is set to the solution x'' and also the new best solution (lines 25-26 of Algorithm 2). Else, the algorithm constructs a new solution using the MX1 crossover operator of Masmoudi et al. (2022a) (lines 27-30 of Algorithm 2). A key feature of the MX1 operator is that the newly constructed solution by crossover will preserve information from the enhanced solution x_{best} identified at the current iteration and another solution by diversification, using our constructive heuristic. This construction mechanism for new solutions facilitates the algorithm to strike the right balance between intensification and exploration.

This process can balance local exploitation and global exploration, facilitating the escape from converging quickly to a local optimum, or converging too slowly with excessive time computational time for a good solution, as faced by most of the population-based metaheuristics (Talbi, 2009).

Algorithm 2: Hybrid Variable Neighborhood Search

23 **Initialize:** neighborhood structures N_h , temperature $T=T_{max}$, $x_{best} = x =$ current initial solution;
ShakingOrder; ImprovementOrder;

24 **Repeat**

25 $OrderN \leftarrow ShakingOrder$ *// initialize the order of the neighborhood*

26 $OrderI \leftarrow ImprovementOrder$ *// initialize the order of the local search*

27 **If** x_{best} is not improved **Then**

```

28      $T := \alpha * T;$  // update the temperature
29 End if
30  $h \leftarrow 1;$ 
31 Repeat
32     Generate a new solution  $x'$  from the  $h^{th}$  neighborhood of  $x$ ;
33     Apply the local search procedure on  $x'$  to obtain  $x''$ ;
34     If  $x''$  better than  $x'$  Or accepted by the acceptance function Then
35          $x = x'';$ 
36          $h \leftarrow 1;$ 
37         For  $i \leftarrow 1$  to  $h_{max}$  Do // Generate the new neighborhood order
38              $l \leftarrow$  neighborhood with high improvement number;
39              $UpdateOrderN(i) \leftarrow l;$ 
40         End For
41          $OrderN \leftarrow UpdateOrderN;$ 
42         For  $j \leftarrow 1$  to  $h_{max}$  Do // Generate the new local search order
43              $l \leftarrow$  local search with high improvement number;
44              $UpdateOrderI(j) \leftarrow l;$ 
45         End For
46          $OrderI \leftarrow UpdateOrderI;$ 
47         If  $(x > x_{best})$  Then
48              $x_{best} = x;$ 
49         Else // construct a new solution
50              $x =$  the current initial solution obtained by the constructive heuristic;
51              $x =$  Crossover  $(x, x_{best});$  //apply OX crossover between current solution and previous  $x_{best}$ ;
52         End Else
53     Else
54          $h \leftarrow h + 1;$ 
55     End If
56 Until  $h \leftarrow h_{max} + 1;$ 
57 Until the number of steps is reached
58 Return:  $x_{best}$ 

```

4.1. Constructive heuristic

The developed constructive heuristic captures several extra features related to visits to water refilling stations and separation sites, compaction of waste, and washing of the bins. The detailed steps of the algorithm are shown in [Algorithm 3](#).

Algorithm 3: Constructive heuristic

1. **Initialize:** List of collection points to be visited B , list of empty routes R , and $CompWastetype_p = \text{Vrai}$;
2. **While** there are unserved collection points in B ;
3. Select randomly a collection point;
4. **If** the current selected collection point can be inserted in the best position in the available route respecting the capacity compartment vehicle and total route duration;
5. **If** the accumulated waste of type p at collection point $i+1 > C_p$ **Then**
6. **If** compaction cost at collection point $i >$ traveled distance cost between current node and separation site **Or** $CompWastetype_p = \text{Vrai}$ **Then**
7. Find the nearest separation site to node i ;
8. Insert separation site;
9. Reset the total available waste of all compartments to zero;
10. Update R ;
11. **For** all waste type p **Do**
12. $CompWastetype_p = \text{Faux}$;
13. **End For**
14. **Else**
15. Activate compaction operation at node i for compartment p ;
16. Reduce the total available waste amount of type p by a compacting factor;
17. $CompWastetype_p = \text{Vrai}$;
18. **End If**
19. **End If**
20. **If** the collection point i must be washed **Then**

```

21.         If the available water amount at  $i \leq$  required amount of water to wash all the bins of  $i$  Then
22.             Wash all bins of the collection point  $i$ ;
23.         Else
24.             Find the water refilling station to node  $i$ ;
25.             Insert water refilling station;
26.             Update  $R$ ;
27.         End If
28.     End If
29.     If  $i$  is not mandatory to be washed and all bins can be cleaned without exceeding maximum
    threshold Then
30.         Clean the bins of the collection point;
31.     End if
32.     Else
33.         A new route is opened;
34.     End If
35. Return:  $R$ 

```

Let B be the list of collection points to be visited, R be initialized to a list of empty routes (vehicles) and $CompWastetype_p$ be initialized to the Boolean *Vrai* which represent the case that the previous operation of waste type $p \in P$ was a compaction mechanism (i.e., the last operation was not visiting a separation site). Solution x is constructed starting from scratch by implementing the following steps. A collection point i (involving different types of waste bins) in list B is selected randomly (line 3 of Algorithm 3) and an attempt is made to insert it into the best position of the route partially constructed so far using an available vehicle, respecting constraints related to compartment capacity, and route duration (line 4 of Algorithm 3). If the accumulated waste of type p at bin $i+1$ exceeds the compartment capacity C^p (line 5 of Algorithm 3), two possibilities are considered. First, the nearest separation site to node i (line 7 of Algorithm 3), is inserted into the current route before servicing the next collection point (line 8 of Algorithm 3). The total available waste of all compartments is then reset to zero (line 9 of Algorithm 3) and all compaction operation of all waste type $p \in P$ are reinitialized to *Faux* (lines 11-13 of Algorithm 3). Second, the compacting operation is activated at node i for compartment p (line 15 of Algorithm 3). Consequently, the total available waste amount is compacted and reduced by a compacting factor (line 16 of Algorithm 3) and the $CompWastetype_p$ is initialized to *Faux*. More specifically, if the compaction cost is greater than the traveled distance cost between the current node and the separation site, then the visit to the separation site takes place. We note that, if the previous operation was a compaction mechanism of the waste p type (and not a visiting of the separation site) (line 6 of Algorithm 3), then we force the vehicle to visit the nearest separation site of node i , and then perform the compacting operation. As mentioned, our problem consists of the operations related to emptying all bins of the set of collection points, cleaning those previous bins (mandatory bins) and the possibility of cleaning no-mandatory bins without exceeding a maximum overflow threshold. To do so, we apply the following steps: at each insertion of the collection point in the partial route, if the collection point is mandatory to be cleaned, then the vehicle must wash the bins of this collection point. If the available amount of water is insufficient to clean this inserted collection point, then the nearest water refilling station to this collection point node i is inserted before visiting this collection point i (lines 20-28 of Algorithm 3). If the current inserted collection point is not mandatory to be washed, all bins of

this collection point cannot be cleaned if the available water is not enough, and also, the maximum accumulated in-cleaned bins do not exceed the maximum threshold (lines 29-31 of [Algorithm 3](#)). We note that if the collection point cannot be accommodated in any of the existing routes of R due to the violation of capacity compartment of the vehicle or the duration route, a new vehicle is assigned to this collection point (line 33 of [Algorithm 3](#)). The following computational steps are applied until all collection points have been inserted to the routes.

In the constructive heuristic, collection points must be placed as optimally as possible. To achieve this, the insertion of a collection point i into a route r is considered. The maximum route duration is a crucial factor, determining if collection point i can be accommodated within route r . This determination involves a forward loop over route r , testing all possible combinations for the insertion position of collection point i . In other words, this process occurs sequentially, starting with the first insertion position in the existing collection points of route r . These stages are repeated for all available routes in the solution. The search can be halted whenever a violation of the total route duration is detected. During this process, all possible insertion positions are identified, and the one with the least reduced cost is selected. Although this approach considers more options, it may lead to a better solution at the expense of increased computational complexity. Similar methods were explored by [Masmoudi et al. \(2017; 2022a\)](#).

4.2. Neighborhood structures

At each step, the HVNS algorithm applies multiple neighborhood search operators in the shaking phase. The neighborhood move is a critical phase that aims to balance the preservation of good solution quality and the perturbation of solutions for explorations ([Hemmelmayr et al., 2009](#)). In our proposed algorithm, four neighborhood search structures – N1, N2, N3, and N4 – with various movements are adopted and presented as follows.

The sequence of the neighborhood search operators is crucial to VNS ([Mladenović and Hansen, 1997](#)). Typically, the neighborhood structures are selected such that the space of the neighborhood grows as VNS proceeds from one structure to the next. At the initial stages $\{N1, N2\}$, mild changes are made near the incumbent solution. More substantial changes in the current solution $\{N3, N4\}$ are performed only if the previous mild changes were ineffective.

To this end, we adopt the following sequence of neighborhood structures – N1, N2, N3, and N4 – to be executed based on the ones proposed by [Masmoudi et al. \(2022b\)](#).

Swap (N1): This neighborhood contains three movies. (1) two collection points randomly selected change their position from a randomly selected route. (2) this move consists of selecting two adjacent collection points randomly selected from two distinct routes and changing their positions. (3) finally, this move is similar to the previous one. However, the difference is that this move consists of selecting a pair of non-adjacent collection points selected randomly from each route selected; then, the order of

swapping these selected collection points is randomly inserted in the sequence of the routes. We note that one move is selected in each iteration. Figure 3 illustrate the different moves of this neighborhood. Let initial routes (routes 1 and 2) of a solution composed of collections points, separation sites ($s=1$ and $s=2$), and a water refilling station ($w=1$) be as follows:

Route 1	0	1	9	3	2	$s=2$	12	$w=1$	10	4	11	$s=1$	0
Route 2	0	6	8	7	5	$s=2$	0						

(1) Swap two collection points 3 and 9 on route 1:

Route 1	0	1	3	9	2	$s=2$	12	$w=1$	10	4	11	$s=1$	0
Route 2	0	6	8	7	5	$s=2$	0						

(2) Swap adjacent collection points from route 1 and route 2 between the collection point 9 and from route 1 and 8 from route 2:

Route 1	0	1	8	3	2	$s=2$	12	$w=1$	10	4	11	$s=1$	0
Route 2	0	6	9	7	5	$s=2$	0						

(3) Swap non-adjacent collection points on route 1 and route 2:

To use the operator, we choose two non-adjacent collection points from each route at random. Suppose that we choose collection points 9 and 4 from route 1, and 6 and 5 from route 2. We swap the locations of collection points 9 and 4 on route 1 and 6 and 5 on route 2. We then place this swap at the start of each route.

Route 1	0	4	9	1	3	2	$s=2$	12	$w=1$	10	11	$s=1$	0
Route 2	0	5	6	8	7	$s=2$	0						

Fig. 3: An example of the three-swap moves of operator N1

Insert (N2): This type of neighborhood consists of two moves. (1) a randomly selected collection point is deleted from a route and inserted in the other position in the same route. (2) this move is similar to the previous one, but the insertion of the removed collection point is inserted to another route. When inserted a collection point, the algorithm verifies all possible combinations of insertions of the selected collection point in the best position by respecting the feasibility of the solution. The algorithm keeps the same position from the deleted route if no feasible insertion is possible. Figure 4 presents an example of the two insertion moves.

Initial Routes:

Route 1	0	1	9	3	2	$s=2$	12	$w=1$	10	4	11	$s=1$	0
---------	---	---	---	---	---	-------	----	-------	----	---	----	-------	---

Route 2	0	6	8	7	5	$s=2$	0
---------	---	---	---	---	---	-------	---

(2) Insertion of the collection point 3 after the collection point 4 on the same route 1

Route 1	0	1	9	2	$s=2$	12	$w=1$	10	4	3	11	$s=1$	0
---------	---	---	---	---	-------	----	-------	----	---	---	----	-------	---

Route 2	0	6	8	7	5	$s=2$	0
---------	---	---	---	---	---	-------	---

(2) Insertion of the collection point 9 from route 1 to route 2

Route 1	0	1	3	2	$s=2$	12	$w=1$	10	4	11	$s=1$	0
---------	---	---	---	---	-------	----	-------	----	---	----	-------	---

Route 2	0	6	8	7	9	5	$s=2$	0
---------	---	---	---	---	---	---	-------	---

Fig.4. An example of the two insertion moves of operator N2

Cross-exchange (N3-N4): In this neighborhood structure, b consecutive collection points are moved from one route (say, Route 1) to another one (say, Route 2) (Osman, 1993). In return, d consecutive collection points are moved from Route 2 to Route 1. b is chosen between 2 (N3) and 3 (N4) randomly, and d is set to b or $b-1$. This procedure effectively offers a higher degree of diversification of the search (Masmoudi et al., 2022b). When the number of collection points to be exchanged is large, the likelihood of providing an effective exchange is low. Thus, the number of collection points to be exchanged is restricted in that interval. To achieve more effective diversification capability, this scheme of the neighborhood move is widely adopted in the perturbation stage (see, e.g., Hemmelmayr et al., 2009; Masmoudi et al., 2022b). To illustrate the key idea of this operator, we provide an illustrative example in Figure 5 for N3, where the same idea can be applied also for N4.

Initial Routes:

Route 1	0	1	9	3	2	$s=2$	12	$w=1$	10	4	11	$s=1$	0
---------	---	---	---	---	---	-------	----	-------	----	---	----	-------	---

Route 2	0	6	8	7	5	$s=2$	0
---------	---	---	---	---	---	-------	---

Assume we chose $b = 2$ and $d = 1$ for the Cross-exchange (N3). First, we choose two consecutive collecting points at random from route 1 and shift them to route 2. For example, we choose collection points 1 and 9 from route 1. Following this move, we get:

Route 1	0	3	2	$s=2$	12	$w=1$	10	4	11	$s=1$	0
---------	---	---	---	-------	----	-------	----	---	----	-------	---

Route 2	0	6	7	1	9	5	$s=2$	0
---------	---	---	---	---	---	---	-------	---

Next, we pick one ($d = 1$) collection site at random from route 2 and shift it to route 1. Assuming we choose collection point 5 from route 2, we obtain the final sequence of this operator N3 after the step of move.

Route 1	0	3	2	5	$s=2$	12	$w=1$	10	4	11	$s=1$	0
Route 2	0	6	7	1	9	$s=2$	0					

Fig.5. An example of cross-exchange N3

4.3. Local search operators

To enhance the solution quality, the HVNS apply four local search operators to improve the quality of the solution, namely, 2-opt (L1) of Lin (1965), the 2-opt* (L2) of Potvin and Rousseau (1993), and the intra- and inter-relocate (L3-L4) operator of Savelsbergh (1992). These operators are applied only on the collections point nodes. These operators must make an assessment of the compartment capacity restriction.

Similar to the adaptive shaking phase, a predefined order of all local searches is considered, and the selection of an appropriate local search for improvement is based on its performance. In our HVNS, we adopt the first improvement strategy. We note that only these operators are considered for the updated order in lines 20-24 of Algorithm2.

It is noted that the N1 and N2 neighborhood search structures, as well as the different node movements for each local search operator, follow the approach described below. At each step, the algorithm checks if the added collection point matches the compartment capacity of the vehicle. Additionally, it checks if the move violates the constraint that all bins at the collection point must be washed. If this is the case, the move is abandoned, and the algorithm continues to search for a possible solution that meets the aforementioned criteria (compartment capacity and need to wash bins).

Since these neighborhood structures and local search operators do not consider compaction operations, separation and water refilling station insertion/removal, there may be instances where certain visits to the separation and water refilling stations and compaction operations are unnecessary. Conversely, some situations may require additional visits to the separation/water refilling or compaction operations to satisfy the constraints. To address these issues, dedicated operators are proposed, as described in the following section.

4.4. Procedures for restoring feasibility

The MC-WCP-BW is a highly complex optimization problem that includes additional constraints such as the need for water refilling, visits to the separation site, and compaction operations to maintain feasibility. The proposed operators have been tailored to the unique characteristics of the MC-WCP-BW and are effective in addressing these constraints. However, further research and experimentation are required to fully realize their potential. The initial results are very promising and highlight the

importance of adapting solutions to the specific constraints of a given problem. The following operators are used at each local search move:

Remove Separation site (RS): This operator was successfully utilized by [Masmoudi et al. \(2022b\)](#) demonstrating an outstanding performance. It is applied to each route of a solution. The key idea of RS is to check if the compartment capacity of the vehicle is respected. Suppose that the vehicle is at a separation facility j , and the next job is to collect bin wastes at node $j+1$. If the total amount of wastes type collected can still be accommodated by the same compartment waste type in the vehicle, then this separation facility node will be removed. An illustrative example of this operator is shown in [Figure 6](#).

Initial Routes:

Route 1	0	1	9	3	2	$s=2$	12	$w=1$	10	4	11	$s=1$	0
Route 2	0	6	8	7	5	$s=2$	0						

The separation $s=2$ from route 1 is deleted. We obtain:

Route 1	0	1	9	3	2	12	$w=1$	10	4	11	$s=1$	0
Route 2	0	6	8	7	5	$s=2$	0					

Fig.6. Remove separation site operator

Remove Compact operation (RC): This operator is very similar to the previous one RS. However, the difference is that this operator is applied for the compaction operation instead of the separation site.

Insert Separation site (IS): IS considers all pairs of nodes (i, j) of each vehicle route ($\forall i, j \in B$) as proposed by [Masmoudi et al. \(2022b\)](#). If the total amount of waste type collected at node j cannot be accommodated by a selected compartment type, then two possibilities can be considered; first, the nearest separation site (defined as with the least detour distance between j and its predecessor) will be inserted. After the vehicle visits a separation site, the total amount of all different waste types collected is set to zero. Second, the vehicle can compact the selected waste type with a factor ϑ^r at node i before going to the next collection point if the distance traveled to the nearest separation site plus the distance from the separation site to the next collection facility is greater than going from the collection point i to j . We note if the vehicle has already compacted the waste of a compartment, then the vehicle should insert the separation site. Then, the vehicle can compact the same waste type again. In other words, at each visiting the corresponding facility, the vehicle can compact the same waste type of this compartment.

Remove Water Station (RWS): This operator is applied at each route to verify the water tank constraint, inspired by [Masmoudi et al. \(2022b\)](#). The role of this operator is to check each pair of nodes (i, j) , where i and j represent two collection points, between which a visit to a water refilling station is scheduled. If the refilling level in the tank water at node i is enough to clean all bins of collection point node j , the water refilling stations node is then deleted.

5555U 3.14 GHz and 8 GB RAM. The MIP solver of CPLEX 12.10.2 is employed to solve the mathematical model through single thread computation.

5.1. Benchmark sets

Since the MC-WCP-BW is a new problem introduced in this paper, no benchmark instances are available in the literature. Therefore, we have generated a new set of instances inspired by the well-known CVRP benchmark instances of [Queiroga et al. \(2021\)](#). The original data set contains 10000 instances. This instance set covers a wide range of problem settings and contains different numbers of clustering customers, variations in customer demand, and different locations of depots. Each instance contains up to 100 customers, where the customer locations are generated in the square area $[0, 1000]$. We remark that, in our case, the customers represent collection points. In our experiment, the minimum subset of these instances, which covers all the characteristics, is extracted from the original data, where different locations of depots (random, site, and corner) and different customer locations (random, clustered, and a mixture of clustered/randomly) are considered. In the original benchmark instances, customer demands (small, large, unitary, etc.) and vehicle capacities (small and large) are considered. To create more challenging instances, we have chosen only the instances where many customer requests and small-capacity vehicles are considered. This selection enforcement that the vehicle compacts the wastes during the route or visits the separation sites due to the large customer demands and small capacity vehicles. Depending on customer locations, these instances can be regrouped into three categories -- A0, A1, and A2. In type A1 instances, locations are distributed randomly, while instances of type A0 have clustered locations, and A2 contains a mixture of clustered/random locations. We note that A0, A1, and A2 contain different locations of depots, demands, and capacities by keeping the same coordinates and values as in the original benchmark instances of [Queiroga et al. \(2021\)](#). The original capacity vehicles and demand values are considered for only one compartment. To adopt these values, we use the same generation idea of [Reed et al. \(2014\)](#) in their MC-WCVRP by slipping the capacity and demand into the number of vehicle compartments. In other words, the vehicle's capacity is divided into four compartments equal to the number of waste types. To complete instances related to the separation and water refilling stations, we apply the following procedure. Regarding separation sites, two and three separation sites are considered as applied in most MC-WCVRP that include the landfill facilities (e.g., [Rabbani et al., 2016; 2017; Farrokhi-Asl et al., 2017](#)). Regarding the water stations, we locate one water refilling station in the depot since refilling at the central depot is a reasonable claim. Moreover, the locations of two other types of water refilling stations are based on the nature of the instance (clustered, randomly, mixture). This procedure is inspired by the traditional WCVRP with a single compartment (e.g., [Kim et al., 2006; Benjamin and Beasley, 2010; Louati, 2016; Tirkolaee et al., 2019](#)), where most of them use two separation sites in each instance. Based on these instances, we have chosen this value, but for water refilling stations which can increase the challenge of the problem and make sure that these

refilling stations are visited. This is also because the water refilling stations are limited (Vieira et al., 2021). The same generation procedure is also applied to the separation locations.

The compaction operation cost is set to 10€ since the compaction operation consumes, on average, five litres of fuel, according to statistical and technical reports (Holmberg et al., 2014; Smart City, 2016; CPHEEO, 2016). The volume of water bins ranges from 120lt, 240lt, 400lt, 660lt, 800lt, to 1100lt. On average, a volume of washing a bin is around 10 litres (Katmerciler company; Tisan company). The capacity of the water tank varies between 400 and 3000 litres depending on the vehicle capacity and the type (Fulongma company; Hidro-Mak company; Tisan company; and Dmix International company). We have chosen a small tank capacity of 400 litres to guarantee that the water refilling stations can be used. Figure 10 provides several capacity values to show their impact on the solution quality and visits to water refilling stations. The full refilling cost of the water vehicle tank at each station is 120€ since the capacity of the water tank is 400 litres (0.3€ per litre). The average operating costs per kilometre (insurance, maintenance, fuel consumption, labour,...etc.) is about 0.69€/km (Levinson et al., 2005). The city center is located at the square of 800*800 in the site of the area. In other words, if $(100 \leq x(i) \leq 900$ and $100 \leq y(i) \leq 900)$, then the collection point i is in the area of the city. The maximum threshold value ψ is set to 10% of the number of collection points outside the city center area. In Table 7 and Figure 5, we provide different combination of the values. The new data sets and the detailed results can be downloaded from <https://mc-wcvrp-wb-36.websselfsite.net/>.

In the following sections, we provide the detailed results of the HVNS after testing it not only on our newly generated MC-WCP-BW instances but also on the MC-MCVRP benchmark instances.

5.2. Setting the parameters

The HVNS has the advantage that it is relatively related to only two parameters, mainly the initial temperature (T_{init}) and the stopping condition of the algorithm. The parameter values are chosen based on the experiments and recommendations conducted in the literature. To drive the simulated annealing procedure slow and smooth, parameter T_{init} is calculated as $T_{init} = \frac{-0.05}{\ln 0.5} * f(x)$, where function $f(x)$ is the cost of the initial solution (Masmoudi et al., 2021). For the stopping criteria, our algorithm returns the best solution when no improvement has been achieved after ten consecutive runs. This stopping criterion is inspired by Masmoudi et al. (2016) and is designed to obtain good-quality solutions. The cooling rate α is equal to 0.99975 as commented by Ropke and Pisinger (2010), Demir et al. (2012) and Masmoudi et al. (2016).

5.3. Comparison on the benchmark instances

To assess the performance of our algorithm, we applied it to the MC-VRP benchmark instances of Abdulkader et al. (2015) that adopted from the well knows instances of Christofdes et al. (1979) where the capacity of the vehicle is split into two compartments (0.25Q and 0.75Q), which can be considered a special case of the MC-WCP-BW. In fact, MC-WCP-BW can be transformed into MC-VRP in which

no separation and water stations are considered, and no vehicles equipped with the compacting mechanism are available.

Figure 8 shows the results of our HVNS on the benchmark instances of Abdulkader et al. (2015) for the MC-VRP where a single visit of a vehicle at the bin is considered. These results are compared with those obtained by the Hybrid self-Adaptive Variable Neighborhood (HAVNS) and Hybrid Artificial Bee Colony (HABC) algorithms of Kaabachi et al. (2019) and the Three-Dimensional Ant Colony Optimization (TDACO) algorithm of Guo et al. (2022), both of which are considered the current state-of-the-art algorithms for solving the MC-VRP large instances. The objective is to minimize the total traveled distance. In Figure 8, we have used the performance profile technique of Dolan and Moré (2002) to assess the performance of each algorithm in each instance. On the y-axis, the % of instances for which a given algorithm has reached a given performance (reported on the x-axis). The performance is the ratio of the objective value obtained by the algorithm to the best known value (for a given instance). The lower the value in the x axis, the better the performance is.

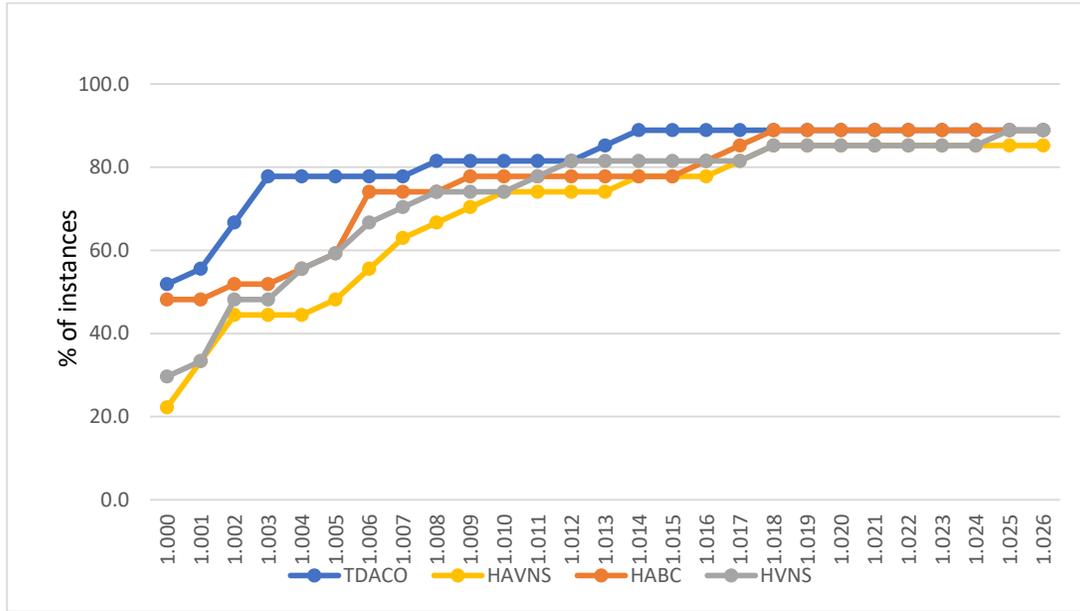


Figure 8: Comparison of our HVNS with the HAVNS, HABC, and the TDACO regarding the total traveled distance

The figure clearly shows that our algorithm compares very well with HABC, and that also with respect to TDACO its maximum deviations are inferior. Our HVNS are competitive with the state-of-art algorithms in terms of quality, albeit with a small difference gap even the HABC and TDACO tailored for the MC-VRP. Figure 8 and the results show that the HVNS has an average deviation of 0.79% from the best-known solutions, while the HAVNS, HABC, and TDACO provide an average deviation of 0.88%, 0.68%, and 1.02%, respectively. This is achieved thanks to the additional diversification and intensification mechanisms that allow converging toward good solutions. Our HVNS reaches the best-known solutions in 7 instances compared to 14 instances for the TDACO and 12 for the

HABC. A trend is remarked regarding the number of best solution results provided by the same metaheuristics approach type, namely our HVNS and the HAVNS of [Kaabachi et al. \(2019\)](#); our HVNS can found 11 better results compared to only eight better results. In addition, our HVNS reaches the best-known solutions in 16 instances compared to 7 instances for the HAVNS. This shows the robustness of our HVNS. Results obtained on the MC-VRP confirm that the proposed HVNS algorithm is competitive with respect to the HAVNS and HABC of [Kaabachi et al. \(2019\)](#) and the TDACO of [Guo et al. \(2022\)](#).

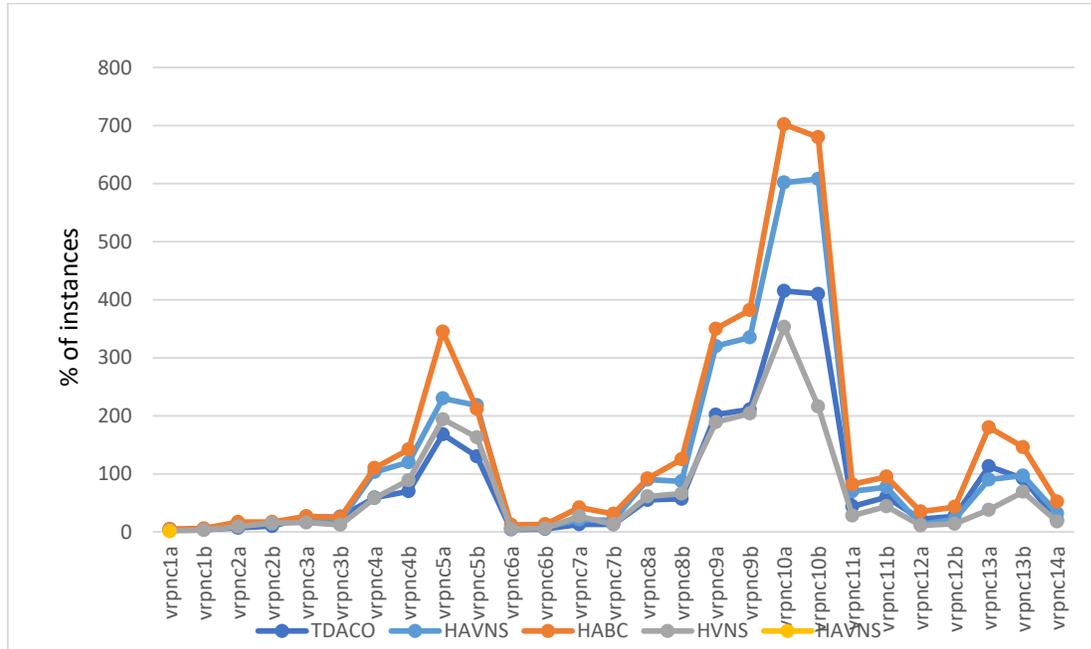


Figure 9: Comparison of our HVNS with the HABC¹ and the TDACO² regarding the computational times

The comparison in [Figure 9](#) and the detailed result on the website indicates that our HVNS clearly outperforms the current state-of-art algorithms in terms of computational times in most instances, where HVNS reports an average equal to 71.22 seconds compared to 120.22 seconds for the HAVNS, 146.96 seconds for the HABC and 83.89 seconds for the TDACO. Thus, our HVNS is considerably faster than the other methods. This is due to the speedup and adaptive intelligence mechanism involved in our HVNS.

5.4. Algorithmic components

In this subsection, we show the impact of the different diversification and intensification procedures, especially the intelligent adaptive mechanisms applied to the local search and shaking phase, the

¹ Results of [Kaabachi et al. \(2019\)](#) executed on 2.20 Ghz Intel Core i5-10310U computer with 6GO of RAM

² Results of [Guo et al. \(2022\)](#) executed on 3.49 Ghz Intel Core computer with 32GB of RAM

crossover operator, and the guided temperature procedure. To this end, new configurations of components are compared to the standard VNS by incorporating different component(s) each time. The detailed results are shown in Table 4, where the MC-VRP benchmark instances of Abdulkader et al. (2015) are used. First, in configuration “C1”, we apply the standard VNS described early in Section 4. Configuration “C2” represents the traditional VNS configuration C1 plus the diversification mechanism with its crossover operator (lines 27-30 of Algorithm 1). In “C3”, we apply our developed guided reducing temperature (lines 5-7 of Algorithm 1) into configuration C1. Configuration “C4” combines different C2 and C3 diversification steps. Configuration “C5” uses the intelligent re-order adaptive mechanism applied to the diversification mechanism (lines 15-18 of Algorithm 1), while Configuration “C6” applies the re-order adaptive mechanism to the intensification mechanism (lines 20-23 of Algorithm 1) instead of using it the diversification phase as in “C2”. We note that both Configurations C5 and C6 are based on Configuration C4. The last configuration, “C7”, reflects our HVNS described in Algorithm 1, in which we use all the components of the diversification mechanism with its crossover operator and guided reducing temperature, the re-ordered adaptive mechanism into the shaking and intensification phases. Table 3 recapitulates these different configurations.

Table 3
A list of different configurations of the algorithmic components

Configuration	Description
C1	Standard VNS
C2	Standard VNS + crossover operator
C3	Standard VNS + reducing temperature procedure
C4	Standard VNS + crossover operator+ reducing temperature procedure
C5	Standard VNS + crossover operator+reducing temperature procedure+adaptive re-ordering neighborhood search
C6	Standard VNS + crossover operator+reducing temperature procedure+adaptive re-ordering local search
C7 (HVNS)	Standard VNS + crossover operator+reducing temperature procedure+adaptive re-ordering neighborhood search +adaptive re-ordering local searches (main algorithm)

Table 4 provides the column “Best%” to represent the deviation gap from the Best-Known Solution (“BKS”).

Table 4
Impact on different components on the solution quality

Inst.	BKS	HABC ^c	TDACO ^f	C1	C2	C3	C4	C5	C6	C7 (HVNS)
		Best (%)	Best (%)	Best (%)	Best (%)	Best (%)	Best (%)	Best (%)	Best (%)	Best (%)
vrpnc1a	550.42 ^c	0.00	0.05	3.51	3.14	2.10	1.25	0.66	0.33	0.05
vrpnc1b	502.83 ^c	0.00	9.14	4.65	4.43	3.09	1.83	0.81	1.68	0.63
vrpnc2a	876.76 ^d	1.59	0.00	2.48	2.09	1.65	1.29	1.86	2.28	1.24
vrpnc2b	888.92 ^d	2.82	0.00	8.75	8.09	7.42	6.51	5.08	4.03	3.59
vrpnc3a	870.80 ^b	0.51	0.00	6.68	4.92	2.82	1.62	0.21	0.81	0.38
vrpnc3b	867.51 ^d	3.21	0.00	4.41	3.12	2.25	1.61	1.02	1.59	1.58
vrpnc4a	1106.84 ^d	1.74	0.00	6.62	5.49	3.08	2.03	1.65	1.06	0.87
vrpnc4b	1126.79 ^d	2.90	0.00	2.94	2.04	2.15	1.77	1.05	1.35	0.89
vrpnc5a	1385.71 ^c	0.00	1.32	5.21	4.62	3.08	1.85	1.43	2.52	0.74
vrpnc5b	1467.39 ^d	1.68	0.00	6.36	5.90	3.57	2.81	3.03	3.33	2.47
vrpnc6a	557.49 ^a	0.00	0.00	3.32	2.31	2.43	1.37	0.00	0.60	0.00
vrpnc6b	503.82 ^c	0.02	10.24	3.36	2.88	2.13	1.79	1.65	1.83	1.18

vfpnc7a	928.24 ^a	0.00	0.21	3.69	2.57	2.78	1.76	1.41	2.73	0.00
vfpnc7b	932.67 ^a	0.00	0.12	3.51	2.60	1.62	1.10	0.18	1.68	0.00
vfpnc8a	880.75 ^d	0.54	0.00	3.15	2.28	2.97	1.94	0.89	1.86	1.03
vfpnc8b	880.89 ^d	0.48	0.00	4.47	2.21	3.08	2.03	1.92	1.78	1.31
vfpnc9a	1211.21 ^c	0.81	0.64	6.33	5.82	3.63	2.03	2.55	1.59	0.41
vfpnc9b	1215.30 ^d	0.88	0.00	9.41	8.64	4.79	2.34	0.85	2.43	0.73
vfpnc10a	1496.56 ^d	0.54	0.00	7.07	4.67	5.87	2.94	0.00	0.72	0.00
vfpnc10b	1508.04 ^d	0.58	0.00	6.80	4.92	3.18	2.27	1.78	1.24	0.98
vfpnc11a	1108.67 ^d	0.16	0.00	6.12	4.13	3.06	2.93	1.93	2.65	1.45
vfpnc11b	1220.43 ^c	0.00	0.28	5.52	4.65	1.91	1.47	1.31	2.58	0.79
vfpnc12a	901.15 ^c	0.00	0.73	4.92	4.52	3.99	2.60	0.77	0.18	0.00
vfpnc12b	923.25 ^c	0.00	3.38	8.04	4.86	5.18	2.23	0.39	1.65	0.00
vfpnc13a	1549.82 ^c	0.00	0.10	5.03	3.87	4.01	2.85	0.30	0.72	0.00
vfpnc13b	1536.38 ^c	0.00	1.24	5.76	5.51	3.44	1.26	1.27	0.93	0.70
vfpnc14a	911.35 ^a	0.00	0.14	7.01	5.58	3.66	3.24	0.00	0.42	0.00
Avg	1033.70	0.68	1.02	5.37	4.29	3.29	2.17	1.26	1.65	0.78

^aResults of *Kaabachi et al. (2019)* executed on 2.20 Ghz Intel Core i5-10310U computer with 6GO of RAM

^aBest known solutions provided by *Abdulkade et al. (2015)*

^b known solutions provided by *Wang et al. (2018)*

^cBest known solutions provided by *Kaabachi et al. (2019)*

^dBest known solutions provided by *Guo et al. (2022)*

^fResults of *Guo et al. (2022)* executed on 3.49 Ghz Intel Core computer with 32GB of RAM

Table 4 shows that applying the traditional VNS (“C1”) cannot provide better results. The average gap to the best-known solutions is equal to 5.37%. After using the crossover operator, we can observe that the quality of solutions has improved, compared to the traditional VNS, with an average gap of 1.08% (between C2 and C1). This can be explained by the good diversification capabilities of crossover in configuration C2. The same observation is obtained using the guided temperature mechanism in the VNS (C3). This improvement is due to the controlled temperature mechanism to explore different search regions.

We observe that, interestingly, using two diversification mechanisms (crossover and controlled reducing temperature in the same combination) in configuration C4 is more effective than using a separate one in each configuration (C2 and C3), with an average gap between C4 and C2 is equal to 2.12% (4.29% - 12.17%). For the case of C4 and C3, the average gap is 1.12% (3.29% - 2.17%). Thus, Configuration C4 shows that using our crossover operator and reducing temperature technique contributes positively to the quality of solutions and outperforms the traditional VNS. However, it still lacks capabilities when it comes to intensification, which the intensification case is clearly shown in C5 and C6. We can observe that using the adaptive re-ordering mechanism is beneficial to enhance the performance of the VNS when applied to configurations C5 and C6 by obtaining improvements for several instances under these configurations. Besides, applying an adaptive mechanism in the shaking phase can improve the quality of the solution (C5) by an average of 1.26%. This effect can also be seen in Configuration C6, where the intelligent re-ordering mechanism is applied in the intensification mechanism that can obtain an average gap equal to 1.65%. In addition, from the detailed result, we can observe that applying this technique in the shaking phase (C5) can significantly improve the quality of the solution. We can find that in several instances, this configuration can reach the best-known solution in three instances. When applying both adaptive mechanisms in the diversification and intensification mechanisms give good results and outperform the average results obtained by the TDACO of *Guo et al.*

(2022), even with a slight improvement of 0.78% for the best-known solutions compared to 1.02% for the TDACO of Guo et al. (2022). However, we find that using the intelligent re-ordering mechanisms both in the intensification and diversification phases has a positive effect on the performance of the VNS method. This enhancement can be attributed to the possible construction of better patterns for solution improvement and the shaking operators, facilitating the exploration of more promising solutions in the search space, as suggested in Karakostas and Sifaleras (2022). In conclusion, applying all added components (adaptive mechanism, crossover operator, and reducing temperature technique) to the traditional VNS is the most effective configuration.

5.6. Experiments with small-size instances

In this section, we provide the results obtained by the HVNS against the one of CPLEX on two data set small-size instances of 10 and 15 bins. These instances are generated as follow. Starting from the large-size instances presented in Section 5.1, we randomly select the number of collection points. Then, we solve the problem by HVNS to assess the results on these instances. Finally, for each type of instances (A0, A1, A2), we choose six instances which are represented by the following convention, e.g., “A0_10_1”. Table 5 reports the objective function values and the computational times obtained executing CPLEX for four hours and also running HVNS five times for each instance. The “UB” column represents the upper bound value, the “LB” column represents the lower bound value reached upon reaching the time limit (bold values indicate instances solved with proven optimality within the time limit), and the percentage gap between the two (Gap%). The “Dev%” column denotes the percentage deviation from the optimal solution, or a lower bound solution obtained by CPLEX. For both CPLEX and the HVNS, the computational time in minutes is reported in column “CPU(min)”.

Table 5
Results obtained by CPLEX and the HVNS on small-size instances

Inst.	CPLEX				HVNS			Inst.	CPLEX				HVNS		
	UB	LB	Gap%	CPU(min)	Dev%	CPU(min)	UB		LB	Gap%	CPU(min)	Dev%	CPU(min)		
A0_10_1	687.86	687.86	0.00	6.71	0.00	0.12	A0_15_1	600.30	600.30	0.00	102.65	0.00	0.45		
A0_10_2	375.23	375.23	0.00	12.69	0.00	0.25	A0_15_2	410.57	410.57	0.00	98.45	0.00	0.26		
A0_10_3	275.05	275.05	0.00	26.72	0.00	0.18	A0_15_3	603.33	599.89	0.57	240.00	0.72	0.98		
A0_10_4	280.49	280.49	0.00	49.92	0.00	0.52	A0_15_4	358.06	358.06	0.00	118.45	0.00	0.63		
A0_10_5	201.37	201.37	0.00	23.30	0.00	0.49	A0_15_5	518.02	518.02	0.00	131.45	0.00	0.57		
A0_10_6	201.72	201.72	0.00	27.78	0.00	0.14	A0_15_6	317.58	317.58	0.00	145.65	1.31	2.60		
A1_10_1	457.94	457.94	0.00	49.95	0.00	1.21	A1_15_1	471.70	460.07	2.47	240.00	0.94	0.87		
A1_10_2	285.57	285.57	0.00	64.21	0.00	0.44	A1_15_2	440.47	431.18	2.11	240.00	0.95	0.70		
A1_10_3	402.53	402.53	0.00	19.81	0.00	1.47	A1_15_3	329.38	325.43	1.20	240.00	0.56	0.92		
A1_10_4	474.13	474.13	0.00	98.46	0.00	0.92	A1_15_4	387.32	378.55	2.27	240.00	0.52	0.42		
A1_10_5	422.48	422.48	0.00	79.45	0.00	1.15	A1_15_5	361.26	349.70	3.20	240.00	0.97	0.94		
A1_10_6	329.09	329.09	0.00	84.50	0.00	0.89	A1_15_6	377.05	365.81	2.98	240.00	1.12	1.72		
A2_10_1	660.07	660.07	0.00	44.15	0.00	0.27	A2_15_1	666.13	649.81	2.45	240.00	0.43	1.12		
A2_10_2	421.61	421.61	0.00	55.65	0.00	0.54	A2_15_2	412.86	412.86	0.00	135.47	0.65	0.66		
A2_10_3	536.97	536.97	0.00	23.45	0.00	1.01	A2_15_3	387.87	380.97	1.78	240.00	0.68	0.74		
A2_10_4	545.69	545.69	0.00	87.14	0.00	0.73	A2_15_4	339.02	339.02	0.00	163.98	0.00	0.82		
A2_10_5	585.45	585.45	0.00	60.65	0.00	0.34	A2_15_5	301.55	298.34	1.07	240.00	0.74	0.43		
A2_10_6	495.94	495.94	0.00	93.40	0.00	1.03	A2_15_6	363.12	356.51	1.82	240.00	0.59	1.04		
Avg	424.40	424.40	0.00	50.44	0.00	0.65	Avg	424.76	419.59	1.22	196.45	0.73	0.88		

The complexity of the problem is assessed in Table 5. While CPLEX can find the optimal solution within the allowed limited time for most instances with ten collection points, after four hours, a significant optimality gap is observed for most instances with 15 collection points. Instances of type A1 and A2, featuring randomized and a mixture of clustered/random locations, respectively, prove particularly challenging to solve. Conversely, for instances of type A0 with clustered locations, CPLEX can find the optimum for most instances in a short time. It's worth noting that our HVNS successfully obtained all optimal solutions found by CPLEX. Moreover, the HVNS can enhance the upper bound obtained by CPLEX in several instances where 15 collection points are considered, with an average gap improvement of 0.73%. All these factors underscore the effectiveness of our developed HVNS. Consequently, CPLEX faces challenges in solving the large-scale problem.

5.7. Results on large MC-WCP-BW instances

In this section, we present the results of our algorithm on the different MC-WCP-BW features. We display the impact of compaction mechanisms and water stations on the solution quality using different scenarios. In addition, we investigate the effect of the threshold value.

5.7.1. Benefit of compacting electric refuse vehicles

In this section, we study the benefits of using the CRVBW. To achieve our goal, an analysis is performed on two scenarios (S1 and S2). The following procedure is applied. Firstly, based on our generated MC-WCP-BW instances described in Section 5.1, the coordinates of the separation nodes are slightly modified. For the first scenario (S1), separation nodes are clustered and located near the urban area as done in the generation location of the collection points in the data set an instance of type ‘‘A0’’, while in the second scenario (S2), separation nodes are located very far from the urban area. Secondly, for each scenario, we evaluate the robustness of using a vehicle with a compactor and a vehicle without a compactor on the solution quality. For a simple notation, we use RVBW (Refuse Vehicle with Bin Washer) and CRVBW (Compressed Refuse Vehicle with Bin Washer).

Table 6 summarizes the results obtained over five runs. In the table, for each scenario, columns Cost% and Dist% give the deviation between the RVBW and CRVBW fleets in terms of costs and distance, respectively.

Table 6
Impact of the compaction operation

Data Set	S1				S2			
	RVBW		CRVBW		RVBW		CRVBW	
	Cost	Dist	Cost%	Dist%	Cost	Dist	Cost%	Dist%
Avg A0	7,679.78	10,110.03	2.20	7.34	8,405.74	11,080.45	-5.57	-4.96
Avg A1	10,057.49	13,544.73	-2.13	-1.22	9,818.39	13,239.73	-8.19	-7.56
Avg A2	14,184.94	19,527.67	0.07	-2.51	15,670.04	21,568.75	-6.38	-3.90

In the first scenario (S1), the RVBWs provide better results than the ones of CRVBWs in all instances of data set ‘‘A0’’ where the separation nodes are clustered, with an average gap equal to 2.20%.

For the data set of type “A1”, we can observe that using a fleet of CRVBWs provides better results compared to the one obtained by the RVBWs with an average gap equal to 2.13% in the data set instances of type “A1” where the location collection points are randomly generated, and the separation locations are clustered. For the data set instances “A2” where the separation sites’ location collection points are clustered and randomly generated points, the results obtained by CRVBWs and RVBWs are comparable, with an average gap equal to 0.08%.

Regarding the second scenario (S2), where the separation locations are located far from the collection points location, we can notice that using CRVBWs is significantly beneficial. It provides better results than using a fleet of RVBWs in data sets A0, A1, and A2 with **exceptions on a few instances, with an average gap equal to 4.96% for data sets “A0”, 7.56% in data set “A1” and 3.90% in data set “A2”**.

In general, regarding the two scenarios, using a fleet of CRVBWs is better than using a fleet of RVBWs in the second scenario, i.e., the CRVBWs are beneficial when the separation sites are located far from the location of the collection points, which leads to the minimization of distance to go through to the separation by collecting more bins. While using a fleet of RVBWs can also be beneficial when the separation and water refilling stations are clustered and near the locations of the collection points, in which the RVBWs can be allowed to be disposed of in the same area of the collection points.

5.7.2 Impact of water tank capacity

In this section, we compare different tank water capacity values, and we show how this could affect the performance of the MC-WCP-BW solution. To do so, we consider the same instances described in [Section 5.1](#). We consider the capacity of the water tank divided by 2.0 and then multiplied by 1.5 and by 2.0. More specifically, if the water tank value is equal to 400 litres, then the test parameters are as follows: 200 litres, 600 litres and 800 litres. [Figure 10](#) shows the impact of the different values on the solution quality. More specifically, [Figure 10.a](#) indicates the percentage of deviation from the best solutions established by using our main water tank value (400 litres). [Figure 10.b](#) gives the average number of visits to the refilling stations for each corresponding water tank capacity, i.e., 200, 400, 600 and 800 litres.

We note that the positive percentage deviations indicate an improvement in solution with respect to the best value found by using our value of 400 litres.

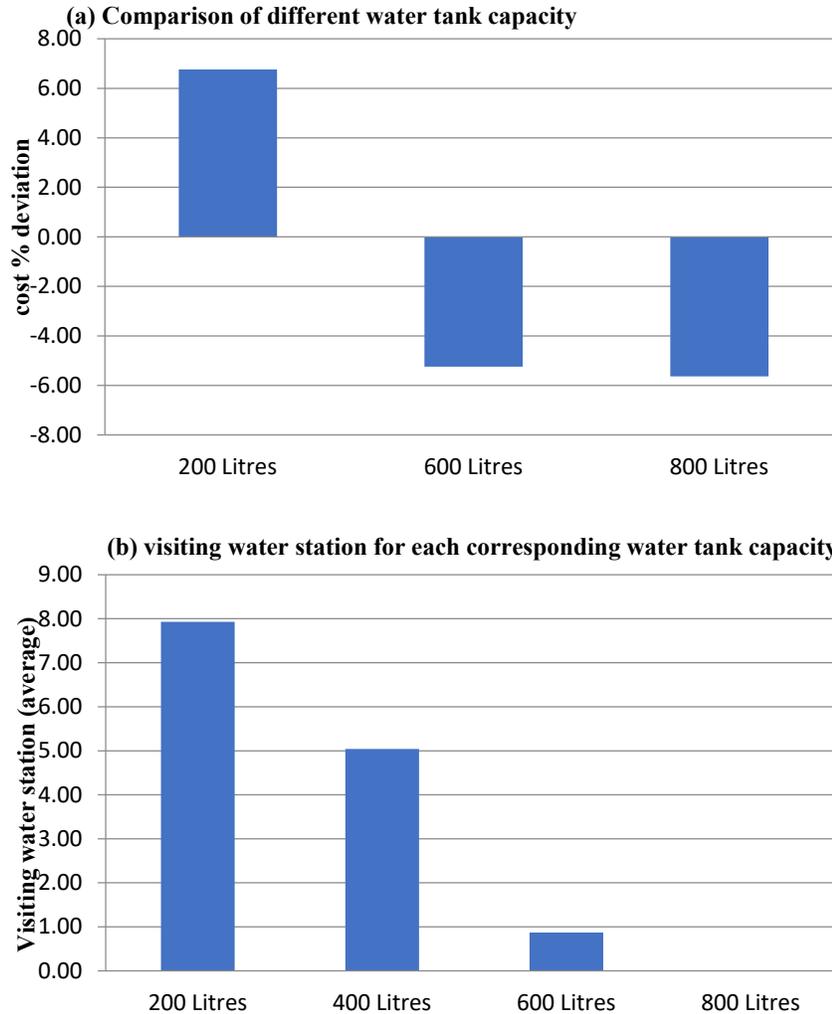


Figure 10: Impact of different capacities of water tanks on the solution quality

From Figure 10, we can observe that using water tank capacity with increased value clearly decreases the total costs due to the reduction of detour distance to visit the water stations. More specifically, using a double water tank capacity value leads to an improvement of 5.63% with respect to the main water tank capacity. In addition, using double capacity value is enough to clean most of the bins compared to the other applied capacity values without visiting any water station. In general, it is clear that by increasing the capacity (600 and 800 litres), the costs and the number of visits to the water stations are reduced. In addition, the settings of using 600 and 800 litres provide similar results with an average gap equal to 5.24% for the capacity tank of 600 litres and 5.63% for the 800 litres with a similar number of visits to the water stations were using the capacity of a tank of 600 litres can wash the bins in several instances without visiting the water stations as using 800 litres. From the detailed results of this table, we notice that in 35 instances among 56 instances, the capacity tank of 600 litres is enough to wash the bins. However, decreasing the capacity tank (200 litres) increases the costs due to the additional distance to the water stations, which is clearly shown in Figure 10.b with an average equal to 7.93 visits

that range between 5 and 11 visits of the water stations. Thus, decreasing and increasing tank capacity impact the water stations visit, detours costs, and the total costs.

5.7.3. Impact of the threshold variation

In this section, we evaluate the impact of the threshold value on the solution quality. We consider different threshold values: 0% (all bins must be washed), 5%, 15%, and 20%. The results in Figure 11a are compared against the main threshold value 10% in terms of total costs, while Figure 11b provides the number of visits with each threshold value of 0%, 5%, 10%, 15%, and 20%.

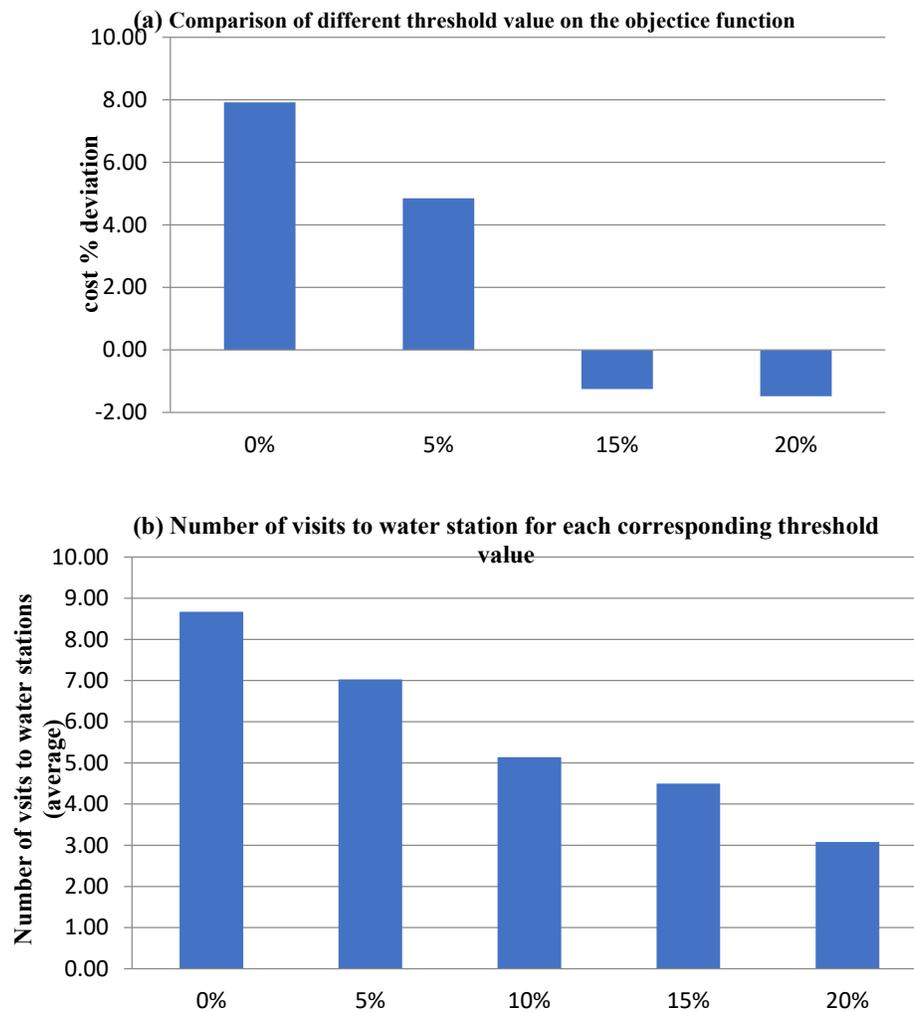


Figure 11: Impact of the threshold value on the number of visits to water stations

From Figure 11a, we observe that when all bins must be washed (0%), an increase of 7.92% in the total costs is obtained compared to the threshold value (10%). In Figure 11b, we also notice that for the case of 0%, the number of visiting water refilling stations is higher than the other threshold values to face the visiting of the water refilling stations with an average equal to 8.67, ranging between 5 and 13. We find that the total cost, in this case, is higher than the other values. Thus, the threshold value is

critical and has an influence on the solution quality. We observe that the total costs are decreased by increasing the threshold value with an average cost equal to 1.25% for the considered 15% and by 1.48% for the case of 20%. In addition, we see that beyond a certain threshold value, a very slightly decreasing cost can be achieved when considering the threshold value 15% and 20% with a different gap of 0.23%, which confirm that the vehicle cannot wash the no- mandatory bins since it is not required to visit the water station. **Similar behavior can be observed in the number of visits to water refilling stations for 10%, 15%, and 20%, with averages of 5.14, 4.50, and 3.08, respectively. In these cases, most visits to water refilling stations fall within the range of 3 to 6 times for the 10%, 2 to 6 times for the 15%, and 1 to 4 times for the 20%.**

5.7.4. Impact of the city center area

The availability of the city center area that imposes the mandatory rule to wash all the bins with the region, plays a critical factor in the solution quality and the washing operations. We conducted experiments using different areas of the city center. We set the following area where the bins must be washed. For a simplified notation, we denote by S1 that the area city center covers a medium part of the whole city which surrounds the square 400*400, and S2, the area city is small that surrounds the square 200*200. Table 7 displays the different obtained results using those tested scenarios S1 and S2 compared to S0, which presents our main area city that covers a very large part of the city. The column “Cost%” provides the percentage gap to the cost obtained by our main scenario S0 where a large city area is considered. Column “Nb_O” represent the average number of collection points located outside the city area.

Table 7
Comparison of different scenarios

Data set	S0		S1		S2	
	Nb_O	Cost	Nb_O	Cost	Nb_O	Cost
Avg A0	15.10	7,369.50	59.10	-4.54	44.10	-0.76
Avg A1	33.04	9,854.91	73.78	-9.36	44.52	-5.50
Avg A2	30.65	12,843.06	56.02	-5.04	42.02	-2.01
Avg	26.27	10,022.49	62.97	-6.31	43.55	-2.76

As expected, using different scenarios values has a considerable impact on the quality of solutions and the number of bins that must be washed. Our results show that using a tight area city in S2 leads to lower costs with more bins outside the city area compared to the largest area city S0 since this gives greater flexibility to choose which bins can be cleaned. Using data set A0, for example, the average gap of the total cost is equal to 4.54% compared to 0.76% for the S2. However, for the same data set type, an increase in the number of bins outside the area site is observed with an average of 43.47 for the S2, 15.10 for the S0, and 59.10 for the S1. Another important observation provided from this table, we can notice that the setting of randomly located customers in the tight area of city S0 results in lower costs with an average gap of 5.50% and with the highest number outside the city area at around 44.52. In contrast, the more clustered collection points with the largest area, city S0 leads to the highest costs with

an average cost compared to all other scenarios and area values equal to 4.54% and with less number outside the city area with an average equal to 15.10. We also notice that our experiments show that using a medium size area city leads to a compromise between reducing the total costs and the number of unwashed bins compared to the other scenarios.

5.7.5. Impact of the re-ordering strategies

In this subsection, we evaluate the impact of using the re-ordering local search and neighborhood search strategies and examine how this could affect the performance of the MC-WCP-BW solution. We are interested in making a comparison between the two algorithms' configurations. For the first one, we apply [Algorithm 2](#) with only re-ordering local search (by applying the traditional neighbourhood search order). For the second one, we apply the same [Algorithm 2](#) by only re-ordering neighborhood search (by applying a random traditional local search order). We denote, by Configuration 1 and configuration 2, the two applied algorithms, respectively. Large-size instances that range between 201 and 501 nodes are used, based on and selected from the large-size CVRP instances proposed by [Uchoa et al. \(2017\)](#), which is similar to the same generation idea of [Queiroga et al. \(2021\)](#). We follow the same strategy applied in [Section 5.1](#) to complete these instances. The results using the two configurations 1 and 2 with our main [Algorithm 2](#) (using both local search and neighborhood search strategies) are given in [Table 8](#). Column “Cost%” indicates the percentage of deviation from the objective function established by using [Algorithm 2](#). The columns “Nb_W%”, “Nb_C%” and “Nb_L%” present the percentage deviation from the number of water stations visited, the number of compaction times, and the number of landfill facilities visited, respectively found by the main [Algorithm 2](#). Finally, the column “Dist%” indicates the percentage gap of the total traveled distances found by each configuration compared to the one found by our main proposed [Algorithm 2](#).

Table 8
Comparison between re-ordering local search strategy and re-ordering neighborhood search strategy

Inst.	Cost	Algorithm 1				Configuration 1					Configuration 2				
		Nb_W	Nb_C	Nb_L	Distance	Costs%	Nb_W%	Nb_C%	Nb_L%	Dist%	Costs%	Nb_W%	Nb_C%	Nb_L%	Dist%
A0_200_1	4097.36	5	14	9	5174.21	1.33	0.00	0.00	0.00	0.00	2.68	0.00	11.26	0.00	2.78
A0_200_2	3182.57	4	15	12	4165.30	6.62	0.00	8.53	0.00	7.32	8.10	0.00	18.09	0.00	8.42
A0_200_3	2609.48	4	9	4	2866.03	7.98	0.00	12.51	0.00	10.37	3.13	0.00	13.12	0.00	3.51
A0_200_4	4844.26	5	12	7	6162.59	1.79	0.00	0.00	0.00	0.00	1.60	0.00	0.00	0.00	0.00
A0_200_5	2368.92	2	12	8	3113.07	4.08	0.00	11.46	0.00	4.00	9.06	0.00	11.91	0.00	9.15
Avg A0	3420.52	4	12	8	4296.24	4.36	0.00	6.50	0.00	4.34	4.91	0.00	10.88	0.00	4.77
A1_200_1	6396.37	7	10	10	8150.81	6.80	0.00	25.20	0.00	7.42	7.87	0.00	13.22	0.00	8.67
A1_200_2	5660.83	6	12	10	6841.25	3.15	0.00	9.96	0.00	3.46	7.16	0.00	10.45	0.00	8.36
A1_200_3	6258.80	2	9	5	8579.06	10.43	1.72	42.13	0.00	10.31	7.57	0.00	29.66	0.00	7.45
A1_200_4	2794.17	2	8	7	3446.32	9.24	0.00	14.15	0.00	9.69	6.46	0.00	30.27	0.00	6.42
A1_200_5	5386.98	6	11	5	6649.55	7.00	0.00	30.89	0.00	7.58	2.09	0.00	10.90	0.00	2.21
Avg A1	5299.43	5	10	7	6733.40	7.32	0.34	24.47	0.00	7.69	6.23	0.00	18.90	0.00	6.62
A2_200_1	5971.12	4	12	5	8209.79	4.64	0.00	20.85	0.00	4.39	8.35	0.00	21.46	0.00	8.23
A2_200_2	11927.34	9	9	6	15628.05	9.86	0.00	55.52	0.00	10.13	3.10	0.00	29.42	0.00	3.24
A2_200_3	4485.03	5	11	4	5541.87	2.77	0.00	0.00	0.00	0.00	6.95	0.00	15.96	0.00	7.90
A2_200_4	5717.15	4	15	11	8042.20	0.00	0.00	0.00	0.00	0.00	3.32	0.00	7.37	1.82	3.45
A2_200_5	9246.95	3	16	5	12591.44	2.83	2.59	14.09	0.00	2.81	0.00	0.00	0.00	0.00	0.00
Avg A2	7469.52	5	12	6	10002.67	4.02	0.52	18.09	0.00	3.47	4.34	0.00	14.84	0.36	4.57

A0_300_1	6166.94	8	22	15	7125.54	1.46	0.00	5.79	0.00	1.54	0.00	0.00	0.00	0.00	0.00
A0_300_2	5996.27	9	19	13	7015.73	1.98	0.00	6.86	0.00	2.14	2.70	0.00	7.02	0.00	3.22
A0_300_3	3626.41	9	13	11	3858.64	0.00	0.00	0.00	0.00	0.00	4.57	0.00	18.37	0.00	5.51
A0_300_4	6292.70	6	23	23	7496.48	1.25	0.00	4.99	0.00	1.25	2.90	0.00	10.17	0.00	2.94
A0_300_5	3843.47	6	12	12	4284.02	5.74	0.00	18.55	0.00	6.69	2.07	0.00	9.54	0.00	2.37
Avg A0	5185.16	8	18	15	5956.08	2.08	0.00	7.24	0.00	2.33	2.45	0.00	9.02	0.00	2.81
A1_300_1	9399.44	6	15	23	12667.29	2.03	0.00	8.26	0.00	2.13	0.98	0.00	8.58	0.00	0.95
A1_300_2	7917.11	11	16	17	9314.99	7.73	0.00	12.63	1.01	9.44	1.00	0.00	6.52	0.00	1.08
A1_300_3	8582.80	8	14	21	11103.99	1.21	0.00	8.55	6.67	1.22	1.24	0.00	8.74	0.00	1.22
A1_300_4	4019.54	5	14	13	4858.52	5.30	0.00	15.92	0.00	6.01	3.39	0.00	8.17	0.00	3.69
A1_300_5	8882.76	11	15	16	10916.64	3.72	0.00	15.90	0.00	4.08	1.75	0.00	8.11	0.00	1.95
Avg A1	7760.33	8	15	18	9772.29	4.00	0.00	12.25	1.54	4.58	1.67	0.00	8.02	0.00	1.78
A2_300_1	7919.57	8	20	13	10372.86	3.83	0.00	6.47	0.00	4.28	2.85	0.00	0.00	0.00	1.05
A2_300_2	13918.58	16	16	25	17136.85	6.80	0.00	21.63	0.00	7.74	1.78	0.00	7.52	0.00	2.00
A2_300_3	4643.00	9	12	14	5565.82	7.98	0.00	16.54	0.00	9.70	7.12	0.00	8.56	1.73	8.93
A2_300_4	7513.90	6	20	16	9830.72	3.52	0.00	17.87	0.00	3.51	7.34	0.00	12.19	0.00	7.77
A2_300_5	17382.90	10	28	25	17998.73	5.97	0.00	3.78	0.00	6.34	4.16	0.00	4.02	0.00	4.53
Avg A2	9555.59	10	19	18	12181.00	5.62	0.00	13.26	0.00	6.31	4.65	0.00	6.46	0.35	4.86
A0_400_1	8324.76	14	26	29	8926.18	2.28	0.00	4.79	0.00	2.74	3.93	0.00	4.97	0.00	4.96
A0_400_2	7663.86	12	21	25	8792.97	3.02	0.00	11.68	0.00	3.37	8.62	0.00	6.03	0.00	10.87
A0_400_3	4751.37	12	16	17	4628.24	6.23	2.54	13.05	6.12	8.51	1.63	0.00	13.66	0.00	1.77
A0_400_4	8498.98	13	28	33	9892.22	2.71	0.00	12.60	0.00	2.88	2.23	0.00	0.00	0.00	2.79
A0_400_5	4452.33	9	18	23	4998.39	5.78	0.00	7.01	0.00	7.02	1.29	0.00	7.29	0.00	1.38
Avg A0	6738.26	12	22	25	7447.60	4.00	0.51	9.83	1.22	4.91	3.54	0.00	6.39	0.00	4.35
A1_400_1	13015.05	13	18	33	16198.93	5.40	0.00	7.18	0.00	6.03	3.12	0.00	7.48	0.00	3.47
A1_400_2	10242.68	12	21	24	12194.30	3.62	0.00	10.18	0.00	4.21	2.59	9.71	10.61	0.00	1.33
A1_400_3	9559.17	15	21	31	11676.77	3.75	3.38	10.77	0.00	4.23	1.79	0.00	5.72	0.00	2.03
A1_400_4	5623.95	10	17	20	6389.16	10.30	11.55	5.95	1.01	10.33	1.25	0.00	6.10	0.00	1.36
A1_400_5	9893.33	14	22	28	11693.76	8.26	0.00	14.92	0.00	9.45	2.19	0.00	5.20	0.00	2.55
Avg A1	9666.84	13	20	27	11630.58	6.26	2.99	9.80	0.20	6.85	2.19	1.94	7.02	0.00	2.15
A2_400_1	10819.08	7	23	20	13622.03	8.55	0.00	10.50	0.00	9.44	10.01	0.00	5.60	0.00	11.21
A2_400_2	16672.69	19	22	28	20821.02	8.26	0.00	13.77	0.00	9.17	7.18	0.00	4.81	0.00	8.62
A2_400_3	8736.85	12	18	17	9715.81	8.36	10.91	15.32	0.00	7.56	6.59	0.00	15.69	1.90	8.08
A2_400_4	9941.94	13	23	24	12195.99	8.22	0.00	13.16	4.39	9.45	3.60	0.00	4.50	2.42	4.05
A2_400_5	17758.13	17	35	39	22265.18	9.61	0.00	9.74	0.00	10.89	1.78	0.00	3.39	0.00	1.91
Avg A2	12785.74	14	24	26	15724.01	8.60	2.18	12.50	0.88	9.30	5.83	0.00	6.80	0.86	6.77
A0_500_1	9502.39	15	29	39	10570.79	6.66	0.00	7.61	0.00	8.41	4.92	0.00	7.96	0.00	5.91
A0_500_2	9275.15	18	28	37	9596.87	6.94	0.00	12.71	0.00	8.72	3.04	0.00	4.48	0.00	4.02
A0_500_3	5443.92	15	21	22	4947.16	7.67	6.26	9.10	4.14	8.00	3.46	2.31	9.61	2.25	4.82
A0_500_4	11189.67	14	36	42	12753.09	7.02	2.84	3.12	0.00	8.60	6.94	8.58	3.34	3.78	6.99
A0_500_5	6405.75	14	23	30	6134.80	3.76	0.00	5.13	0.00	5.24	2.52	0.00	5.26	0.00	3.45
Avg A0	8363.38	15	27	34	8800.54	6.41	1.82	7.53	0.83	7.79	4.17	2.18	6.13	1.21	5.04
A1_500_1	16177.76	16	21	38	19842.87	5.21	0.00	11.75	2.93	5.87	5.23	0.00	6.06	0.00	6.01
A1_500_2	13801.70	19	28	32	15566.46	6.44	6.25	12.90	0.00	6.13	1.06	0.00	4.47	0.00	1.22
A1_500_3	12353.72	14	22	46	14644.88	4.41	8.04	14.77	0.00	3.52	1.09	0.00	5.15	0.00	1.16
A1_500_4	6561.55	14	24	26	6859.71	3.23	0.00	9.13	3.65	3.98	1.96	0.00	4.89	4.55	2.46
A1_500_5	13698.98	22	24	41	14934.90	5.98	0.00	9.14	5.48	7.63	2.34	0.00	14.08	1.59	2.56
Avg A1	12518.74	17	24	37	14369.76	5.05	2.86	11.54	2.41	5.43	2.34	0.00	6.93	1.23	2.68
A2_500_1	11893.01	14	26	28	14450.49	8.54	0.00	7.85	0.00	9.85	0.96	0.00	4.16	0.00	1.03
A2_500_2	18827.98	24	28	45	22781.97	6.66	3.37	3.48	0.00	7.81	0.86	0.00	3.72	0.00	0.95
A2_500_3	8313.91	16	20	25	8921.24	5.29	6.73	4.84	0.00	4.79	1.51	0.00	10.04	7.41	1.68
A2_500_4	14597.06	15	33	35	17084.15	6.45	0.00	10.93	0.00	7.34	2.89	3.34	3.89	1.75	3.31
A2_500_5	23125.05	17	38	41	29859.58	4.07	6.87	5.76	5.34	3.71	3.56	7.50	3.03	0.00	3.35
Avg A2	15351.40	17	29	35	18619.49	6.20	3.39	6.57	1.07	6.70	1.96	2.17	4.97	1.83	2.06
Avg	8676.24	11	19	21	10461.14	5.33	1.22	11.63	0.68	5.81	3.69	0.52	8.86	0.49	4.04

From Table 8, we observe that using both strategies in the same framework leads to better solutions compared to using separate ones, as clearly shown in all instances where positive percentage values are obtained in both configurations 1 and 2 compared to the main algorithm. It is shown that by considering intelligent re-ordering mechanisms both in the intensification and diversification phases, there is a

positive impact on the performance of HVNS. This enhancement may be attributed to the potential construction of better patterns of improvement and shaking operators, facilitating the exploration of more promising solutions in the search space, as well as by utilizing longer execution time for the most successful operators, as suggested by [Karakostas and Sifaleras \(2022\)](#).

In addition, we can see that using Configuration 1 where re-ordering local search is applied, provides better results where the customers are clustered (A0), and the number of customers ranges between 200 and 300, compared to Configuration 2, with an average gap of 4.36% compared to 4.91% for the case of 200 customers and with 2.08% compared to 2.45% for Configuration 2 where 300 customers are considered. This is due to the less traveled distance and the number of times the vehicles compact the wastes, which is indicated in the difference. However, the configuration provides better results in other data set types for most large-size instances. More specifically, significant improvements in instances with up to 400 customers and 500 customers are observed. In addition, we can observe the instances of type A1 where the number of customers with up to 500 provides the high difference costs compared to the other ones with an average difference equal to 2.71% (5.05%-2.34%). This is relative since we also note that in some instances, the number of water stations visited, and the number of landfill stations visited are higher (indicated in bold) than the ones found by Configuration 2 (in instances with up to 300 to 500 customers). Thus, the different features and components of this problem highly impact solution quality, especially in large-size instances.

5.8. Comparison between the HVNS with respect the initial solutions

In order to evaluate the efficiency of the HVNS algorithm, we compare its solutions to those obtained by the constructive heuristic approach described in [Section 4.1](#). To this end, we select five cases from each dataset A0, A1, and A2 for each number of collection points, each with varying features such as vehicle compartment capacity, depot location, and waste demands at collection points. We run the HVNS algorithm five times on each instance and calculate the percentage improvement over the initial solution. The results are presented in [Figure 12](#).

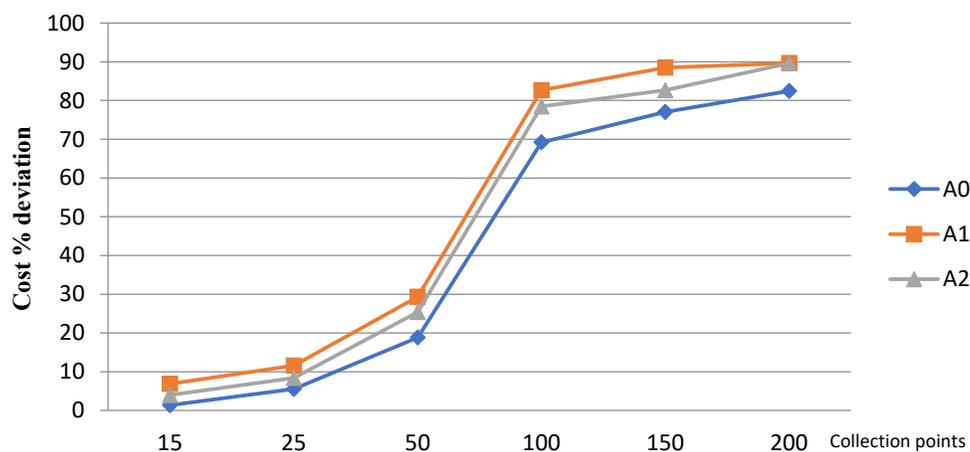
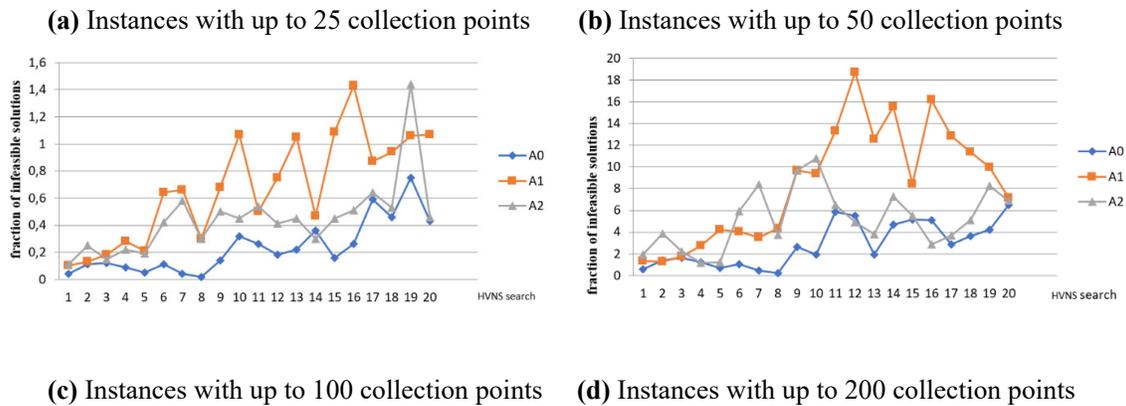


Fig.12. Percentage improvement of the HVNS over the initial solutions

The results presented in Figure 12 demonstrate that the HVNS significantly improves the solutions obtained by the constructive heuristic, with an average improvement of 89% over the initial solutions. As expected, the average improvement increases with the number of collection points. For small instances (15-25 collection points), the improvement is around 11%, with most cases showing an improvement of less than 8%. Interestingly, the results also indicate that the data set A0, which includes instances with 15 collection points clustered around the separation and water refilling stations, provides the least improvement, with an average gap of only 1.37%. This suggests that the constructive heuristic approach is effective in providing good initial solutions for this type of instance with a small number of collection points. Additionally, it is observed that the A0 data set provides less improvement on average compared to A1 and A2, likely due to the randomly located collection points, separation, and water refilling stations in the latter two data sets. Finally, as the number of collection points increases towards 100, the improvement becomes less significant for all data set types.

5.9. HVNS with the infeasible solutions

This section examines the effectiveness of the operators utilized in our approach, including the technique for repairing infeasible solutions that violate the maximum route duration constraint described in Section 4.3. Specifically, we evaluate the impact of different neighborhood search structures, local search operators, and repair operators in terms of compaction operation, insertion/removal of separation and water refilling stations on the ability to repair infeasible solutions. We use the same subset of instances from the previous subsection. Figure 13 illustrates the fraction of infeasible solutions encountered during the HVNS for different instance classes after employing various neighborhood search structures (N1-N4) and local search operators (2-opt, 2-opt*, and intra- and inter-relocate) as well as repair operators (RC, IS, RWS, and IWR). In Figure 13, we divide the total number of algorithm iterations into 20 equal-sized buckets and recorded the fraction of infeasible solutions in each bucket after applying the aforementioned techniques.



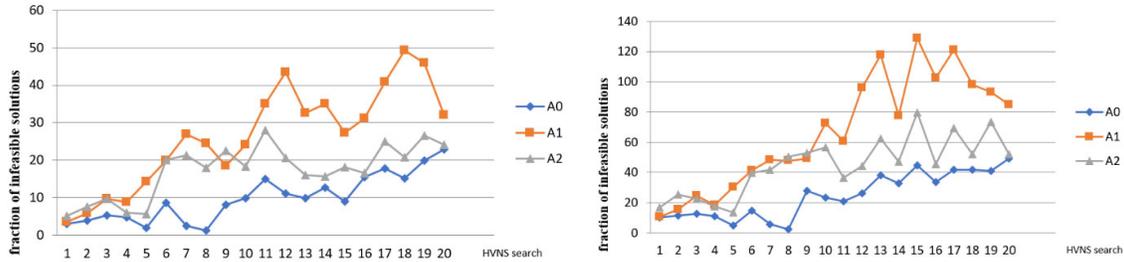


Fig.13. Fraction of infeasible solutions encountered during the HVNS search

The time spent by the algorithm to reach an infeasible solution can vary greatly depending on the instance type. However, as the search progresses, the number of infeasible solutions identified decreases for all instance types. In particular, finding infeasible solutions for A1 instances appears to be more challenging than for A0 or A2 instances, where the collection points are clustered or randomly placed, respectively. This may be attributed to the fact that the collection points, separation and water refilling stations are randomly positioned and far apart in A1 instances.

However, it is worth noting that using A0 instances does not pose such an infeasible solution problem compared to the other instance types. This is because the separation/water refilling station is located close to the collection points, resulting in a short detour distance. Moreover, for clustered instances, finding efficient routes without repairing infeasible solutions tends to result in smaller total route time violations than for instances with randomly placed collection points. This highlights the further benefits of the additional operators for removing/inserting water refilling stations and separation in cases where the collection points, separation, and water refilling stations are clustered.

The proportion of infeasible solutions tends to increase as the number of collection points grows, especially for type A1 instances, where repairing the solution with respect to the maximum route time violation is typically necessary. However, the detailed results in [Figure 13](#) demonstrate that for most examples with 15 collection points in the A0, A1, and A2 datasets, the fraction of infeasible solutions is less than 0.5%. These findings suggest that the combination of various operators for the infeasible separation sites, water refilling stations, and compaction operations with the repairing infeasible solution approach for the maximum route duration is beneficial and highlights the effectiveness of our algorithm.

5.10. Impact of the different repairing infeasible solution components

Figure 13 demonstrates the effectiveness of our repairing method across various instance types. We further conducted a sensitivity analysis to examine the impact of repairing different infeasible solution components, specifically focusing on the restoration of the algorithm's search with a new solution based on the constructive heuristic (Line 50 Algorithm 2). To achieve this, we compared the repair technique outlined in this section with the well-known strategic oscillation method introduced by Glover (2000). This method allows infeasible solutions until the search discovers a high-quality feasible solution for instance type A1, where up to 50 bins are randomly located. The details of the evaluation strategy can

be found in the Appendix. In Figure 14, we illustrate the progress of the search for feasible solutions. The figure presents the results of both procedures during the iterations and the percentage gap associated with the best solution found in this instance.

The concept of accepting infeasible solution spaces with the expectation of improving on them in subsequent iterations is the foundation of strategic oscillation. It accepts such an infeasible solution by penalizing its objective function. Based on Cordeau et al. (1997) and Toth and Vigo (2003), we assign the following evaluation function to each solution, which is written as follows: $f(x) = c(x) + \sum_{p \in P} \alpha_p d_p(x) + \beta z(x) + \gamma w(x) + \tau a(x) + o(x)$. The term $c(x)$ gives the objective function (1) of solution x expressed in Section 3.

Moreover, the terms $d_p(x)$, $z(x)$, and $w(x)$, represent the load, duration, and washing bins violations, respectively. The violations are calculated as follows: $d_p(x) = \sum_{i=1}^n (u_i^p - C_p)^+$, $z(x) = (B_n - B_0 - T_{max})^+$, and, $w(x) = \sum_{i=1}^n (o_i - Q)^+$. The associated penalty parameters α_p , β , and γ are dynamically adjusted during the search (as in Cordeau et al. (1997) and Toth and Vigo (2003), and Sadati and Çatay (2021)). We note that a solution x can only become a new solution if $d_p(x) = z(x) = w(x) = 0$.

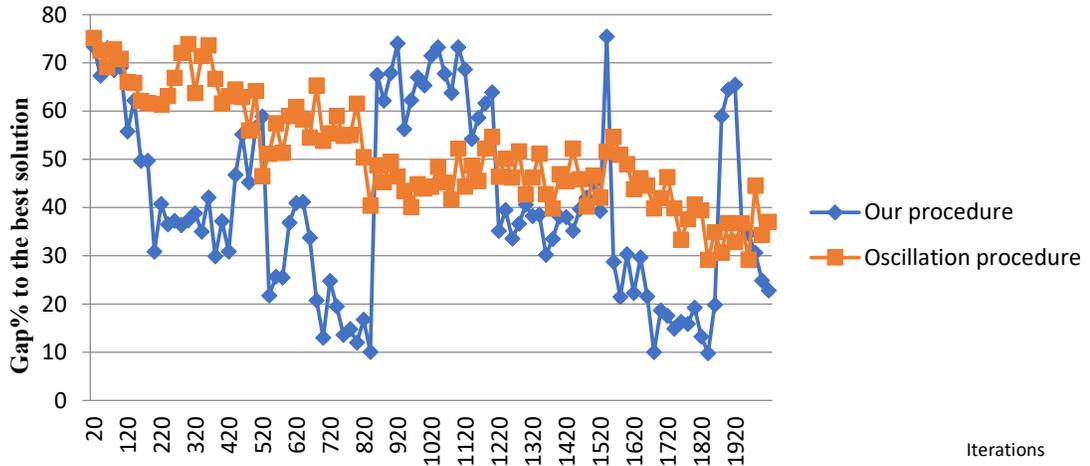


Fig.14. Comparison between our repairing infeasible solution and the oscillation procedure

Figure 14 demonstrates the effectiveness of our method, which proves to be much more efficient than oscillation in the majority of repetitions. The findings highlight that oscillation strategies require more time and iterations to converge. When comparing the oscillation technique to our new search and convergence to a feasible and good solution, we can better observe the impact of developing a new solution as part of our process for fixing infeasible solutions. Finding effective solutions appears to be more challenging, as indicated in Figure 13. This phenomenon could be attributed to the vehicles taking lengthier diversions and violating the maximum route duration.

Another observation is that our procedure works similarly to a multi-start technique, as described by [Masmoudi et al. \(2022\)](#). To explore alternative search areas, our technique for repairing infeasible solutions resumes after a few consecutive iterations from a different starting point. Restarting the search from an unexplored region in the search space aims to increase diversity, enabling the algorithm to avoid getting stuck in ineffective iterations at local optima. This approach helps the algorithm efficiently identify effective solutions. Furthermore, the impact of restarting the search with a new solution during the process of repairing infeasible solutions is evident in how quickly the algorithm converges to good results after generating each new solution, as illustrated in different iterations in Figure 14. The figure clearly demonstrates that the proposed algorithm outperforms the oscillation procedure.

6. Managerial Insights

The computational study discussed in Section 5 provides valuable insights for decision-makers. One significant finding, presented in Section 5.7.2, is the impact of water tank capacity on operating costs. Our study reveals that a higher tank capacity results in lower operating costs due to fewer detours. However, this increase in capacity incurs a higher fixed cost for a larger tank. Our model provides a systematic approach to assess these trade-offs and evaluate the optimal water tank capacity.

Another noteworthy finding, discussed in Section 5.7.3, is the impact of bin-washing requirements on costs. Our study finds that by relaxing the percentage of bins requiring washing, significant cost savings could be achieved. However, this poses a question for governments on whether they should allocate more operating costs for frequent washing to maintain city hygiene. Our methodology offers a quantitative tool to evaluate the cost-effectiveness of different hygiene policies and empower governments to make informed decisions.

Finally, our study shows in Section 5.7.4 that the density of the city center area has a significant impact on waste management and washing operations' quality. We find that in more densely populated areas, waste collection operating costs could be reduced, and there is greater flexibility in choosing which bins to clean. On the other hand, more clustered collection points in large cities could lead to higher costs. Therefore, waste management decision-makers must carefully evaluate the trade-offs between costs and the number of unwashed bins when selecting the city center area for waste management operations.

7. Conclusions

This paper studies a new variant of the Multi-Compartment Waste Collection Routing Problem (MC-WCVRP), which considers a fleet of compressed refuse vehicles equipped with bin washers. The vehicles can perform compaction operations for different types of waste and washing operations for the bins. The bin washers can be refilled at water refilling stations, and the waste of the bins can be emptied at separation sites. We call the problem the Multi-Compartment Waste Collection Problem with Bin Washer (MC-WCP-BW).

We describe a novel mathematical formulation for the MC-WCP-BW, where a multigraph is used to model vehicle routes, thus reducing the number of decision variables. To solve large-scale MC-WCP-BW instances, we design a HVNS that incorporates a novel double adaptive mechanism to obtain high-quality solutions and several diversification mechanisms.

The mathematical formulation and the HVNS algorithm are extensively tested on benchmark sets taken from the literature and on newly generated instances designed explicitly for the MC-WCP-BW. The results obtained show that the proposed HVNS offers good computational performance when compared with the state-of-the-art algorithms for the MC-WCVRP, namely Hybrid self-Adaptive Variable Neighborhood (HAVNS) and Hybrid Artificial Bee Colony (HABC) algorithms of Kaabachi et al. (2019) and the Three-Dimensional Ant Colony Optimization (TDACO) algorithm of Guo et al. (2022). Moreover, experiments are conducted to attest to the effectiveness of the different algorithm components, and managerial insights are also given based on a set of sensitivity analyses on key parameters of the problem: (i) the benefit of carrying out the compaction operation, (ii) the impact of the capacity of the water tank and the variation of threshold on the solution quality, and (iii) the impact of having different dimensions of the city center area on the number of bins that need to be washed.

Our study also offers a number of interesting managerial insights, for example, the impacts of the water tank capacity, the bin-washing requirements, and the density of the bins at the city center. All of these different settings illustrate the trade-offs of the different waste management performance metrics. In order to optimize waste management operations, decision-makers must determine the importance of the costs and the hygiene conditions. This requires a careful evaluation of the trade-offs between these factors to determine the most effective approach to managing waste in the city.

Finally, real-world variants of the problem considered in our paper pose several additional challenges, such as different specialized recycling facilities and multiple service zones, as studied by Anderluh et al. (2021). Our future work will consider these extensions and other important features of this class of problems. Future research could investigate how the theoretical concepts examined in this study could be implemented in real-world contexts to provide practical solutions to existing problems. Moreover, future research could investigate the consequences of allowing many consecutive compaction operations of the same waste type along the same route, such as fuel consumption and hydraulic system requirements. It would be necessary to define parameters to account for previously compacted garbage and its influence on subsequent compaction. The investigation might also examine the efficiency and long-term effects of permitting consecutive compactions without visiting a separation site.

Acknowledgments

The research of the last author was partially supported by the Research Grants Council (RGC) (Theme-based Research Scheme T32-707-22-N), the 2019 Guangdong Special Support Talent Program – Innovation and Entrepreneurship Leading Team (China) (2019BT02S593), and National Natural

Science Foundation of China (72231002). The authors would like to thank the anonymous reviewers and area editor for their valuable comments and thorough evaluation of the manuscript.

References

- Abdulkader, M. M., Gajpal, Y., & ElMekkawy, T. Y. (2015). Hybridized ant colony algorithm for the multi compartment vehicle routing problem. *Applied Soft Computing*, 37, 196-203.
- Anderluh, A., Nolz, P. C., Hemmelmayr, V. C., & Crainic, T. G. (2021). Multi-objective optimization of a two-echelon vehicle routing problem with vehicle synchronization and ‘grey zone’ customers arising in urban logistics. *European Journal of Operational Research*, 289(3), 940-958.
- Benjamin, A. M., & Beasley, J. E. (2010). Metaheuristics for the waste collection vehicle routing problem with time windows, driver rest period and multiple separation facilities. *Computers & Operations Research*, 37(12), 2270-2280.
- Bing, X., Bloemhof, J. M., Ramos, T. R. P., Barbosa-Povoa, A. P., Wong, C. Y., & van der Vorst, J. G. (2016). Research challenges in municipal solid waste logistics management. *Waste management*, 48, 584-592.
- Bruglieri, M., Mancini, S., & Pisacane, O. (2019). More efficient formulations and valid inequalities for the green vehicle routing problem. *Transportation Research Part C: Emerging Technologies*, 105, 283-296.
- Central Public Health and Environmental Engineering Organisation (CPHEEO), 2016. Municipal Solid Waste Management. Available on <http://mohua.gov.in/upload/uploadfiles/files/Part2.pdf>. Last accessed on 27/01/2023.
- Christiansen, M., Fagerholt, K., Rachaniotis, N. P., & Stålhane, M. (2017). Operational planning of routes and schedules for a fleet of fuel supply vessels. *Transportation Research Part E: Logistics and Transportation Review*, 105, 163-175.
- Christofides PTN, Mingozzi A (1979) The vehicle routing problem, In: Combinatorial optimization. Wiley, Chichester, pp 315–338
- Chen, J., Dan, B., & Shi, J. (2020). A variable neighborhood search approach for the multi-compartment vehicle routing problem with time windows considering carbon emission. *Journal of Cleaner Production*, 277, 123932.
- Cordeau, J. F., Gendreau, M., & Laporte, G. (1997). A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks: An International Journal*, 30(2), 105-119.
- Demir, E., Bektaş, T., & Laporte, G. (2012). An adaptive large neighborhood search heuristic for the pollution-routing problem. *European journal of operational research*, 223(2), 346-359.
- Dmix International compagny. Available on COMPACTOR-22-mc-with-bin-washer.pdf (dmixinternational.com). Last accessed on 27/01/2023.
- Dolan, E. D., & Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical programming*, 91(2), 201-213.
- Elbek, M., & Wøhlk, S. (2016). A variable neighborhood search for the multi-period collection of recyclable materials. *European Journal of Operational Research*, 249(2), 540-550.
- Erdem, M. (2022). Optimisation of sustainable urban recycling waste collection and routing with heterogeneous electric vehicles. *Sustainable Cities and Society*, 103785.
- Erdoğan, S., & Miller-Hooks, E. (2012). A green vehicle routing problem. *Transportation research part E: logistics and transportation review*, 48(1), 100-114.
- Farrokhi-Asl, H., Tavakkoli-Moghaddam, R., Asgarian, B., & Sangari, E. (2017). Metaheuristics for a bi-objective location-routing-problem in waste collection management. *Journal of Industrial and Production Engineering*, 34(4), 239-252.
- Feng, J., Xu, S. X., Xu, G., & Cheng, H. (2022). An integrated decision-making method for locating parking sites of recyclable waste transportation vehicles. *Transportation Research Part E: Logistics and Transportation Review*, 157, 102569.
- Fulongma company. FULONGMA pure electric garbage compactor truck function, configuration, and highlights. Available on <https://www.fulongmagroup.com/news/fulongma-pure-electric-garbage-compactor-truck-function-configuration-and-highlights/>. Last accessed on 27/01/2023.
- Gajpal, Y., Abdulkader, M. M. S., Zhang, S., & Appadoo, S. S. (2017). Optimizing garbage collection vehicle routing problem with alternative fuel-powered vehicles. *Optimization*, 66(11), 1851-1862.
- Guo, N., Qian, B., Na, J., Hu, R., & Mao, J. L. (2022). A three-dimensional ant colony optimization algorithm for multi-compartment vehicle routing problem considering carbon emissions. *Applied Soft Computing*, 127, 109326.
- Helesi compagny. Available on <https://www.helesi.com/product.aspx?Id=53>. Last accessed on 27/01/2023.
- Hemmelmayr, V. C., Doerner, K. F., & Hartl, R. F. (2009). A variable neighborhood search heuristic for periodic routing problems. *European Journal of Operational Research*, 195(3), 791-802.

- Henke, T., Speranza, M. G., & Wäscher, G. (2015). The multi-compartment vehicle routing problem with flexible compartment sizes. *European Journal of Operational Research*, 246(3), 730-743.
- Henke, T., Speranza, M. G., & Wäscher, G. (2019). A branch-and-cut algorithm for the multi-compartment vehicle routing problem with flexible compartment sizes. *Annals of Operations Research*, 275(2), 321-338.
- Hidro-Mak compagny. Available on <https://pdf.directindustry.com/pdf/hidromak/rear-loaded-garbage-truck-body-container-washing-system/180350-900749.html>. Last accessed on 27/01/2023.
- Hiermann, G., Hartl, R. F., Puchinger, J., & Vidal, T. (2019). Routing a mix of conventional, plug-in hybrid, and electric vehicles. *European Journal of Operational Research*, 272(1), 235-248.
- Hiermann, G., Puchinger, J., Ropke, S., & Hartl, R. F. (2016). The electric fleet size and mix vehicle routing problem with time windows and recharging stations. *European Journal of Operational Research*, 252(3), 995-1018.
- Hyva company. Available on <https://www.hyva.com/fr-ma/>. Last accessed on 27/01/2023.
- Holmberg, K., Andersson, P., Nylund, N. O., Mäkelä, K., & Erdemir, A. (2014). Global energy consumption due to friction in vehicles and buses. *Tribology International*, 78, 94-114.
- Hsu, Y. C., Walteros, J. L., & Batta, R. (2020). Solving the petroleum replenishment and routing problem with variable demands and time windows. *Annals of Operations Research*, 294(1), 9-46.
- Kaabachi, I., Yahyaoui, H., Krichen, S., & Dekdouk, A. (2019). Measuring and evaluating hybrid metaheuristics for solving the multi-compartment vehicle routing problem. *Measurement*, 141, 407-419.
- Katmerciler company. Available on <http://www.katmerciler.com.tr/L/EN/mid/343/g/0/c/3/id/51/Refuse-Container-Washing-&-Disinfecting-Vehicle.htm>. Last accessed on 27/01/2023.
- Karakostas, P., & Sifaleras, A. (2022). A double-adaptive general variable neighborhood search algorithm for the solution of the traveling salesman problem. *Applied Soft Computing*, 121, 108746.
- Khajouei, M. H. S. H., Lotfi, M., Ebrahimi, A., & Jafari, S. (2020, December). Water truck routing optimization in open pit mines using the general algebraic modelling system approach. In *International Conference on Logistics and Supply Chain Management* (pp. 255-270). Springer, Cham.
- Kiilerich, L., & Wöhlk, S. (2018). New large-scale data instances for CARP and new variations of CARP. *INFOR: Information Systems and Operational Research*, 56(1), 1-32.
- Kim, B. I., Kim, S., & Sahoo, S. (2006). Waste collection vehicle routing problem with time windows. *Computers & Operations Research*, 33(12), 3624-3642.
- Lahyani, R., Coelho, L. C., Khemakhem, M., Laporte, G., & Semet, F. (2015). A multi-compartment vehicle routing problem arising in the collection of olive oil in Tunisia. *Omega*, 51, 1-10.
- Landfill Alternative company. Available on <https://landfillalternatives.co.uk/static-vs-portable-waste-compactors/>. Last accessed on 27/01/2023.
- Levinson, D. M., Corbett, M. J., & Hashami, M. (2005). Operating costs for trucks. Available at SSRN 1736159.
- Lin, S. (1965). Computer solutions of the traveling salesman problem. *Bell System Technical Journal*, 44(10), 2245-2269.
- Louati, A. (2016). Modeling municipal solid waste collection: A generalized vehicle routing model with multiple transfer stations, gather sites and inhomogeneous vehicles in time windows. *Waste Management*, 52, 34-49.
- Masmoudi, M. A., Braekers, K., Masmoudi, M., & Dammak, A. (2017). A hybrid genetic algorithm for the heterogeneous dial-a-ride problem. *Computers & operations research*, 81, 1-13.
- Masmoudi, M. A., Hosny, M., & Koç, Ç. (2021). The fleet size and mix vehicle routing problem with synchronized visits. *Transportation Letters*, 1-19.
- Masmoudi, M. A., Hosny, M., Braekers, K., & Dammak, A. (2016). Three effective metaheuristics to solve the multi-depot multi-trip heterogeneous dial-a-ride problem. *Transportation Research Part E: Logistics and Transportation Review*, 96, 60-80.
- Masmoudi, M. A., Mancini, S., Baldacci, R., & Kuo, Y. H. (2022a). Vehicle routing problems with drones equipped with multi-package payload compartments. *Transportation Research Part E: Logistics and Transportation Review*, 164, 102757.
- Masmoudi, M. A., Coelho, L. C., & Demir, E. (2022b). Plug-in hybrid electric refuse vehicle routing problem for waste collection. *Transportation Research Part E: Logistics and Transportation Review*, 166, 102875.
- Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. *Computers & operations research*, 24(11), 1097-1100.
- Mofid-Nakhaee, E., & Barzinpour, F. (2019). A multi-compartment capacitated arc routing problem with intermediate facilities for solid waste collection using hybrid adaptive large neighborhood search and whale algorithm. *Waste Management & Research*, 37(1), 38-47.
- Molina, J. C., Eguia, I., & Racero, J. (2019). Reducing pollutant emissions in a waste collection vehicle routing problem using a variable neighborhood tabu search algorithm: a case study. *Top*, 27(2), 253-287.
- Muyldermans, L., & Pang, G. (2010). On the benefits of co-collection: Experiments with a multi-compartment vehicle routing algorithm. *European Journal of Operational Research*, 206(1), 93-103.

- Nolz, P. C., Absi, N., Feillet, D., & Seragiotto, C. (2022). The consistent electric-Vehicle routing problem with backhauls and charging management. *European Journal of Operational Research*, 302(2), 700-716.
- Oliveira, A. D., Ramos, T. R. P., & Martins, A. L. (2015). Planning collection routes with multi-compartment vehicles. In *Operations Research and Big Data* (pp. 131-138). Springer, Cham.
- Orakçi compagny. Available on <https://orakcimakina.com/refuse-compactor-with-bin-washer-obw/>. Last accessed on 27/01/2023.
- Osman, I. H. (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of operations research*, 41(4), 421-451.
- Ostermeier, M., Henke, T., Hübner, A., & Wäscher, G. (2021). Multi-compartment vehicle routing problems: State-of-the-art, modeling framework and future directions. *European Journal of Operational Research*, 292(3), 799-817.
- Pelletier, S., Jabali, O., & Laporte, G. (2019). The electric vehicle routing problem with energy consumption uncertainty. *Transportation Research Part B: Methodological*, 126, 225-255.
- Pisinger, D., & Ropke, S. (2010). Large neighborhood search. In *Handbook of metaheuristics* (pp. 399-419). Springer, Boston, MA.
- Potvin, J. Y., & Rousseau, J. M. (1993). A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *European Journal of Operational Research*, 66(3), 331-340.
- ProCompactor. Available on <https://www.procompactor.com/refuse-garbage-compactor-truck/>. Last accessed on 27/01/2023.
- Queiroga, E., Sadykov, R., Uchoa, E., & Vidal, T. (2021, November). 10,000 optimal CVRP solutions for testing machine learning based heuristics. In *AAAI-22 Workshop on Machine Learning for Operations Research (ML4OR)*.
- Rabbani, M., Farrokhi-Asl, H., & Asgarian, B. (2017). Solving a bi-objective location routing problem by a NSGA-II combined with clustering approach: application in waste collection problem. *Journal of Industrial Engineering International*, 13(1), 13-27.
- Rabbani, M., Farrokhi-asl, H., & Rafiei, H. (2016). A hybrid genetic algorithm for waste collection problem by heterogeneous fleet of vehicles with multiple separated compartments. *Journal of Intelligent & Fuzzy Systems*, 30(3), 1817-1830.
- Rafco company. Available on. <https://www.rafcocommunicapal.com/kopyas%C4%B1-spress>. Last accessed on 27/01/2023.
- Ramos, T. R. P., de Morais, C. S., & Barbosa-Povoa, A. P. (2018). The smart waste collection routing problem: Alternative operational management approaches. *Expert Systems with Applications*, 103, 146-158.
- Reed, M., Yiannakou, A., & Evering, R. (2014). An ant colony algorithm for the multi-compartment vehicle routing problem. *Applied Soft Computing*, 15, 169-176.
- Sadati, M. E. H., & Çatay, B. (2021). A hybrid variable neighborhood search approach for the multi-depot green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 149, 102293.
- Savelsbergh, M. W. (1992). The vehicle routing problem with time windows: Minimizing route duration. *ORSA journal on computing*, 4(2), 146-154.
- Schneider, M., Stenger, A., & Goeke, D. (2014). The electric vehicle-routing problem with time windows and recharging stations. *Transportation science*, 48(4), 500-520.
- Sethanan, K., & Pitakaso, R. (2016). Differential evolution algorithms for scheduling raw milk transportation. *Computers and Electronics in Agriculture*, 121, 245-259.
- Siswanto, N., Wiratno, S. E., Rusdiansyah, A., & Sarker, R. (2019). Maritime inventory routing problem with multiple time windows. *Journal of Industrial & Management Optimization*, 15(3), 1185.
- Smart City, 2016. First fully electric-powered waste collection vehicle. Available on <https://smartcity.wien.gv.at/site/en/electic-powered-waste-collection-vehicle/>. Last accessed on 27/01/2023.
- Talbi, E. G. (2009). *Metaheuristics: from design to implementation*. John Wiley & Sons.
- Tasar, B., Türsel Eliiyi, D., & Kandiller, L. (2019). Vehicle routing with compartments under product incompatibility constraints. *Promet-Traffic&Transportation*, 31(1), 25-36.
- Tirkolaee, E. B., Alinaghian, M., Hosseinabadi, A. A. R., Sasi, M. B., & Sangaiah, A. K. (2019). An improved ant colony optimization for the multi-trip capacitated arc routing problem. *Computers & Electrical Engineering*, 77, 457-470.
- Tisan compagny. Available on https://www.tisan.net/en/waste_combi_waste_bin_collection_and_compaction_equipped_with_bin_washer.html. Last accessed on 27/01/2023.
- Tiwari, M. K., Kumar, S., & Shankar, R. (2006). Solving part-type selection and operation allocation problems in an FMS: An approach using constraints-based fast simulated annealing algorithm. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 36(6), 1170-1184.
- Uchoa, E., Pecin, D., Pessoa, A., Poggi, M., Vidal, T., & Subramanian, A. (2017). New benchmark instances for the capacitated vehicle routing problem. *European Journal of Operational Research*, 257(3), 845-858.

- UNEP - United Nations Environment Programme (2013). Municipal solid waste: Is it garbage or gold? Available at: https://na.unep.net/geas/getUNEPPageWithArticleIDScript.php?article_id=105. Last accessed on 15/07/2022.
- Urli, T., & Kilby, P. (2017, August). Constraint-based fleet design optimisation for multi-compartment split-delivery rich vehicle routing. In *International Conference on Principles and Practice of Constraint Programming* (pp. 414-430). Springer, Cham.
- Vieira, Y. E. M., de Mello Bandeira, R. A., & da Silva Júnior, O. S. (2021). Multi-depot vehicle routing problem for large scale disaster relief in drought scenarios: The case of the Brazilian northeast region. *International Journal of Disaster Risk Reduction*, 58, 102193.
- Wang, C. L., & Li, S. W. (2018). Hybrid fruit fly optimization algorithm for solving multi-compartment vehicle routing problem in intelligent logistics. *Advances in Production Engineering & Management*, 13(4), 466.
- Yahyaoui, H., Kaabachi, I., Krichen, S., & Dekdouk, A. (2020). Two metaheuristic approaches for solving the multi-compartment vehicle routing problem. *Operational Research*, 20(4), 2085-2108.
- Yousra, B., & Ahmed, E. A. (2022, June). Multi-compartment vehicle routing problem for waste collection in smart cities. In 2022 IEEE 6th International Conference on Logistics Operations Management (GOL) (pp. 1- 7). IEEE.
- Zbib, H., & Laporte, G. (2020). The commodity-split multi-compartment capacitated arc routing problem. *Computers & Operations Research*, 122, 104994.
- Zhang, S., Mu, D., & Wang, C. (2020). A Solution for the Full-Load Collection Vehicle Routing Problem With Multiple Trips and Demands: An Application in Beijing. *IEEE Access*, 8, 89381-89394.