Received: 6 May 2021 | Accepted: 6 May 2021

DOI: 10.1002/net.22061

SPECIAL ISSUE ARTICLE

WILEY

Grocery distribution plans in urban networks with street crossing penalties

Carmine Cerrone[1] | Raffaele Cerulli[2] | Anna Sciomachen[1]

[1]Department of Economics and Business Studies, University of Genoa, Genoa, Italy
[2]Department of Mathematics, University of Salerno, Fisciano SA, Italy

Correspondence
Anna Sciomachen, Department of Economics and Business Studies, University of Genoa, Genoa, Italy.
Email: sciomach@economia.unige.it

Abstract

Following the emergency caused by the Covid-19 pandemic, there is the need, among other measures, to modify urban mobility plans in order to reduce the use of collective public transport, reducing the crowding of people while also preventing traffic congestion through discouraging the use of private vehicles. From this perspective, retail companies operating within cities must also reorganize themselves, considering both the unpredictable requirements of environmental sustainability and the new mobility needs calling for the promotion of bicycles and electric scooters. In this context, we deal with the need to determine minimum cost routes in urban areas for delivering orders placed through e-channels. More precisely, we face a variant of the green vehicle routing problem of heterogeneous fleets, in which the objective function includes environmental impact cost components that differ by vehicle type. Moreover, as a novel issue, attention must be paid to avoid crossing and passing close to bicycle lanes; therefore, penalties are associated with the transit of vehicles near bicycle lanes. To address this problem, we propose a mixed integer linear programming model and a matheuristic associated with it. The proposed approach is then used to analyze different scenarios derived from the transportation network of the city of Milan, Italy. Milan is one of the smartest cities in Europe from the mobility point of view but also one of the most affected by the Covid-19 pandemic, and the municipality is making a big investment to promote the use of bicycles.

KEYWORDS

delivery problem, matheuristic, mixed integer linear programming, urban networks

1 | INTRODUCTION

The emergency caused by the Covid-19 pandemic at the beginning of 2020 is imposing new rules on society at the level of both individual and collective behavior. Cities are now forced to modify their mobility plans to consider, among other issues, new social distancing needs. In fact, in spring 2020, one of the effects observed in large cities immediately after the so-called first lockdown phase and during the temporary resumption of production activity was the propensity for more individual rather than collective mobility, with a strong focus on environmental sustainability. Great economic support has been directed toward this transportation paradigm. For example, the European economic community has made efforts to promote sustainable mobility in urban areas, taking into account the strong limitations that have necessarily been imposed on collective public transport, such as metro and city buses, which now have a lower maximum number of passengers allowed on board. From the citizens' point of view, the trend toward individual mobility is due to a certain reluctance to use collective urban transport for fear of being infected by Covid-19. At the same time, at the level of local administrators, there is concern that the increase in the use of private transport, that is, cars, may lead to high traffic situations and consequently cause excessive delays in home-work transport times. For this reason, in many countries including Italy, local initiatives now encourage the use of bicycles and scooters, both electric and nonelectric. However, it is noted that at the end of the first emergency phase caused by Covid-19, when the resumption of production and commercial activity was very partial and therefore traffic was very limited, bicycles were indeed a very safe means of transport in urban areas. Now, however, there is concern that with the imminent reopening, even if partial, of economic activities as well as schools, universities, etc., the increase in traffic will certainly increase the risk for cyclists. Indeed, in many cities it is not always possible to circulate in urban areas while keeping to the cycle paths, and there are many dangers for cyclists, including intersections, roads with rails, and bumpy sidewalks.

In this scenario, large-scale retail companies and B2C organizations in general that operate within cities are experiencing a period of great difficulty. They are challenged to determine distribution plans that will supply their customers while also meeting both environmental sustainability requirements and the new mobility rules discussed above. Therefore, particular attention must be paid to finding delivery routes that minimize environmental impact, that is, external costs due to pollution, climate change and accidents, while paying attention to two-wheeled traffic, trying to intersect cycle lanes as little as possible and avoid passing near them. Yet since minimizing cost is one of the most important and well-known objectives for logistics companies, it is still not quite clear how such companies can minimize the costs of emissions impacts. This is despite the fact that much attention has been given to this topic, usually classified as the green vehicle routing problem (GVRP) [12], in the recent operations research and management science literature. Lin et al. [15] provide a classification of the GVRP into several subtypes: the green-VRP, the pollution routing problem (PRP) and the VRP in reverse logistics. Yavaz [20] presents a variant of the GVRP with a homogeneous fleet of green vehicles. Further, some authors have recently focused on evaluating environmental impacts on optimal routes. In particular, Ehmke et al. [11] compare the total cost of a route with respect to different components, such as driving cost and emissions cost, and analyze the impact of each component in the search for the optimal solution. Cerulli et al. [9] determine distribution plans for a B2C company that include negative externalities due to emissions of the vehicle used, and they show that better results, in terms of both business and environmental performance, are obtained if environmental issues are included in the distribution model from the beginning. Behnke and Kirschstein [4] consider different origin–destination paths with respect to their emissions and calculate the possible emission savings depending on the road type. Zhang et al. [21] analyze a VRP in many different cities, considering different congestion charge hypotheses.

However, to the best of our knowledge, no studies in the recent literature examine how the incidence of urban bicycle flow can affect the determination of routes for logistic services. One related effort comes from Masłowskia et al. [17], who give an overview of existing routing algorithms when different road traffic conditions and urban policies are considered. In addition, a survey by Asghari et al. [3] contains a systematic and very up-to-date analysis of the literature on GVRP.

To meet this need, this paper focuses on determining the fleet and the routes of nonhomogeneous vehicles of a B2C company that serves customers within a city. We propose a variant of the GVRP in which the objective function includes external costs related to pollutant emissions and which penalizes the transit of commercial vehicles along roads running parallel to or intersecting cycle lanes. The company's vehicles differ from each other in capacity and environmental impact. Further, as a novel issue, attention is paid to the increasing traffic due to bicycles and scooters. More precisely, in order to minimize intersections and proximity bicycle lanes along the new routes, penalties are associated with some arcs of the underlying network models representing urban streets where bicycles circulate. The idea is to analyze whether and how these street crossing penalties promote the choice of alternative routes compared to those more naturally chosen when only the classic shortest path is considered. This problem thus bears some similarity to the routing problem with penalties proposed in Bräysy et al. [5], where the authors use exact models and existing heuristics to optimally solve capacitated VRP instances when turning penalties are assigned. Yet in contrast, the penalties proposed in the present paper vary according to both the street that is crossed and the type of vehicle driving the route. At least to the authors' knowledge, this is the first time that this type of penalty is considered in the objective function of a GVRP problem aimed at minimizing the cost components, which include length, traveling, pollutant emission costs, and urban streets to cross. To address this problem, we propose an approach that combines the classic max-saving algorithm [10, 19] hybridized with the carousel greedy (CG, [8]) and an integer programming optimization. As a blend of metaheuristic and exact approaches, this method is usually called matheuristic [16].

CG is a technique capable of improving the performance of greedy algorithms, successfully used in conjunction with other heuristic and metaheuristic approaches (see References [7, 14]). The proposed approach can be classified as a "one-shot approach" (see [2]); in fact, to set variables in order to execute the MILP model only once using a reduced input, we use a novel metaheuristic algorithm based on a combination of max-saving-CG.

The paper is organized as follows. In Section 2 we present in detail the urban distribution problem under consideration along with its related network model. Section 3 gives the proposed MILP model, while Section 4 presents the resolution approach, based on a suitable modification of the max-saving algorithm and a matheuristic procedure. Section 5 presents the computational experimentation performed on some instances involving the distribution plans of a large-scale B2C company operating in the city of Milan (Italy), where a huge investment is being made to increase cycle paths in preparation for increased demand after Covid-19 emergency. Finally, we give some conclusions and direction for future work.

2 | THE DISTRIBUTION PROBLEM AND THE RELATED URBAN NETWORK MODEL

The problem we focused on can be seen as a particular case of the GVRP, originating from the need to deliver grocery goods ordered via e-channels. In particular, the problem lies in determining the fleet and the routes of nonhomogeneous vehicles to be used for delivering, at minimum cost, the required goods to customers located in urban areas. More formally, the minimum cost routes have to be determined within the transportation network of the city where the delivery service is carried out. Then, as a first step, we represent the urban transportation network as a digraph $G = (N, A)$ where the set of nodes N contains the depot d (or node 0), that is, the set contains n customers and also the origin and destination nodes of each route. Let C denote the set of customers to be served. $N = C \cup \{d\}$ then has $n + 1$ nodes. The arcs of set A are the shortest paths between each pair of nodes of N, computed along the streets of the city on the basis of the travel distance (in km) between nodes i and j, $\forall i, j \in N$ weighted with the unit cost coefficient described below. Note that in this way, no transit nodes are considered in our network model. Hence, let l_{ij} be the weight of arc $(i,j) \in A$ expressing the shortest distance from node i to node j. Therefore, we have to determine a set of routes originating and terminating at a depot and serving all customers, and we also must determine which vehicles to select for traveling along the routes. Let us then have associated with each customer $i \in C$ the value q_{ip}, which represents the required units of goods, expressed in boxes, and the type of products p. Since we are dealing with delivery plans for grocery goods, note that a product p can be fresh, dry or a combination of both.

As in any VRP formulation, each node can only be served once by one vehicle. Concerning the vehicles, they differ by weight class, Euro class, size, capacity and their ability to serve customer $i \in N$ with product p. Hence, let V be the set of all available vehicles of v different types. We assume that a maximum number m_v of vehicles of type v is available for the delivery service.

A feasible vehicle route $\rho = \{d, c_1, c_2, \dots, c_{l-1}, c_l, d\}$ of length l is an ordered sequence of different customers to be served such that the total capacity of the vehicle is not exceeded and all the constraints are respected. A feasible solution $S = \{\rho_1, \rho_2, \dots, \rho_k\}$ of the problem is a collection of feasible routes.

The goal is to minimize the total cost, which includes traveling, loading and pollutant cost components, depending on the type of the chosen vehicle.

Further, as explained in Section 1, we associate a penalty δ_{ijv} with each arc $(i,j) \in A$ with the aim of avoiding the crossing of bicycle lanes in the routes. More precisely, as we will explain in more detail in Section 5, the values δ_{ijv} depend on the vehicle type v, the road between node i and node j, and the proximity of the road to bicycle lanes and intersections. The total unit cost c_{ijv} per km for a vehicle of type v, $\forall v \in V$, traveling along arc $(i,j) \in A$, is then given in (1).

$$c_{ijv} = (t_v + e_v) \cdot \delta_{ijv}, \tag{1}$$

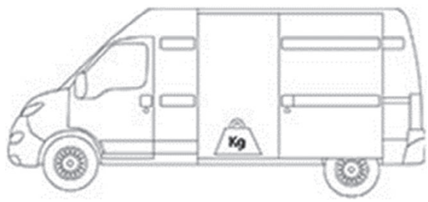
where t_v is the unit traveling cost and e_v is the pollutant emission cost. We denote by $c(\rho)$ the total cost of route ρ and by $c(S) = \sum_{\rho_i \in S} c(\rho_i)$ the total cost of a feasible solution S. Our GVRP problem consists in computing the minimum cost set $S = \{\rho_1, \rho_2, \dots, \rho_k\}$ of feasible routes such that all the customers are served and each customer is visited by a single vehicle that is suitable for delivering the type of product they require.

3 | THE MILP MODEL

In this section we present the proposed MILP model. We first present the relevant notations:

- d depot of the delivery network, origin and destination node of the route;
- c_{ijv} unit traveling cost of arc $(i,j) \in A$ of vehicle of type v (defined in (1));
- e_v unit emission class cost of vehicle of type v;
- t_v unit traveling cost of vehicle of type v;
- w_v weight class of vehicle of type v;
- m_v maximum number of vehicles of type v available for delivery;
- l_{ij} length of arc $(i,j) \in A$;
- δ_{ijv} street crossing penalty, depending on arc $(i,j) \in A$ and the type of vehicle v;
- k_v capacity of vehicle of type v (maximum number of boxes);
- q_{ip} demand of product p (expressed in number of boxes) of customer $i \in C$;
- q_i $= \sum_p q_{ip}$ total demand of customer $i \in C$.

TABLE 1 Characteristics of the vehicles

	Vehicle type	Weight (t)	Euro class	Size (l · w · h) (m)	Capacity (Boxes)	Authorized to deliver		
						Fresh	Dry	Mix
1		—	—	0.5 · 0.5 · 0.5	3	x	x	
2		≤ 1.5	V-VI	4.5 · 2 · 1.8	36		x	
3		≤ 3	V-VI	6 · 2 · 2.5	135	x	x	x

The considered decision variables of the B2C distribution problem under study are the following:

x_{ijv} binary variable, assuming value 1 if a vehicle of type v travels from i to j, and 0 otherwise;
y_{ijv} continuous variable, representing the amount of goods carried by a vehicle of type v from i to j; $y_{ijv} \in [0, k_v]$.

Note that we decided not to generate, instead of setting to zero, each variable x_{ijv} or y_{ijv} corresponding to an arc $(i,j) \in A$, which connects a customer whose demand q_{ip} relates to products that cannot be delivered by a vehicle of type v (see Table 1).

To consider the green and penalty components of our problem, we modified the definition of the costs associated with the arcs of the digraph, as given in (1); therefore, the following model is formally identical to the classic VRP model.

$$\min \sum_{v \in V} \sum_{(i,j) \in A} c_{ijv} \cdot l_{ij} \cdot x_{ijv}, \tag{2}$$

s.t.

$$\sum_{v \in V} \sum_{i \in N} x_{ijv} = 1 \qquad \forall j \in C, \tag{3}$$

$$\sum_{i \in N} x_{ikv} - \sum_{j \in N} x_{kjv} = 0 \qquad \forall k \in N \backslash \{d\}, \forall v \in V, \tag{4}$$

$$\sum_{i \in N} y_{ikv} - \sum_{j \in N} y_{kjv} = q_k x_{ikv} \qquad \forall k \in N \backslash \{d\}, \forall v \in V, \tag{5}$$

$$y_{ijv} \leq (k_v - q_i) x_{ijv} \qquad \forall (i,j) \in A, \forall v \in V, \tag{6}$$

$$y_{ijv} \geq q_j x_{ijv} \qquad \forall (i,j) \in A, \forall v \in V, \tag{7}$$

$$\sum_{j \in C} x_{djv} \leq m_v \qquad \forall v \in V, \tag{8}$$

$$x_{ijv} \in \{0,1\} \qquad \forall i,j \in N, \forall v \in V. \tag{9}$$

$$y_{ijv} \geq 0 \qquad \forall (i,j) \in A, \forall v \in V. \tag{10}$$

In the above formulation, the objective function (2) minimizes the total cost to serve all customers with the chosen fleet of vehicles. Note that value c_{ijv} is computed according to the unit cost per km associated with the pollutant emissions and weight class of vehicle v and the street crossing penalty δ_{ijv}, depending on both the street and the vehicle. This unit cost is then multiplied by the length l_{ij} of arc $(i,j) \in A$. Constraint (3) specifies that a customer is visited exactly once, while constraint (4) specifies that if a vehicle visits a customer, it must also depart from them. Constraint (5) is the so-called flow-commodity constraint, specifying the difference between the quantity of goods a vehicle carries before and after visiting a node. Constraint (6) ensures that the capacity of the selected vehicle is never exceeded. Constraint (7) guarantees that the value y_{ijv} must be at least equal to q_j if a vehicle v crosses the arc $(i,j) \in$ A. Constraint (8) fixes the maximum number of vehicles of each type v that may leave the depot. Finally, (9) and (10) define the decision variables.

4 | THE RESOLUTION APPROACH

In this section we describe in detail the main components of the proposed matheuristic. It has been designed as a resolution technique by combining the well-known max-saving (MS) heuristic [10, 19] with the MILP model, presented in functions (2)–(10) in the previous section. In particular, we suitably modified the MS heuristics to solve the GVRP related to the grocery distribution problem under analysis.

A technique often used to apply a matheuristic is to partition the problem into smaller subproblems through the use of any heuristic procedure [1]. The generated subproblems are then solved by the corresponding mathematical model. Eventually, the solutions of these subproblems are used to obtain a solution of the original problem. In a recent paper [13], the authors proposed this kind of matheuristic approach to solve a problem arising in the management of a bike-sharing system. In our approach, in contrast, the MS heuristic is first modified in order to obtain not just one but a very large number of feasible solutions. We then use the modified heuristic to determine which arcs of the graph have a high likelihood of being part of a good feasible solution and, at the same time, which arcs have a high probability of being discarded. More precisely, the goal of our proposed matheuristic is to check if there is, in the considered urban transportation network G, any type of arc that is very often either excluded or selected in the solutions found by MS heuristics. In particular, we are confident that this approach prioritizes selection of the arcs with lower values of the penalty δ_{ijv}, thus modifying what would be the optimal path, which would ordinarily have the minimum distance as its objective function. Note that a feasible solution is always ensured by MS heuristics. However, in the design of our matheuristic we will carry out a careful tuning analysis of the various parameters involved in the instance. Thus we will highlight the relationship (if any) between the various parameters associated with the arcs and the variations of the solutions produced.

This information about the probability of each arc being selected will be used to appropriately reduce the number of arcs of graph G by removing those arcs with low probability of being selected in the optimal solution. Consequently, the model will have an easier instance to solve. In some sense, we are looking for the "core" of the problem and are interested in narrowing down the potential solutions as much as possible.

4.1 | MS implementation and CG technique

In this subsection we describe in detail the modified MS procedure used to generate many different solutions (not necessarily feasible) that we will use to "reduce" the size of the instances to be solved by model (2)–(10). It is worth observing that the MS procedure we implemented is not intended to create feasible solutions, but only to create sets of routes useful for suggesting the set of arcs, that is variables, to be considered in the MILP model. The procedure suggests to us which arcs of G to remove and which arcs we should confidently choose (and so it also suggests which model variables should be, a priori, set equal to 0 and which variables should be set equal to 1).

The whole procedure is as follows. We execute the MS procedure by selecting each type v of vehicle one at a time, and we create a number of routes equal to the number of customers that can be served by a vehicle of type v. The MS solution is represented by a set of routes S such that $|S| \leq |C|$. These mini routes, which each serve only one customer, will be of the type d-j-d for each served customer $j \in C$. Then, for each pair of routes, say ρ_1 and ρ_2, the savings resulting from both the ρ_1-ρ_2 merger and the ρ_2–ρ_1 merger are calculated (see Figure 1); this is done by either queuing the customers of ρ_2 to ρ_1 or by queuing the customers of ρ_1 to ρ_2. If there are no positive savings, the procedure ends. Finally, all possible savings are sorted in nonascending order and, from these, a random savings belonging to the first 25% of the ordered savings is chosen. The route merger associated with this last savings is applied.

To improve the solutions provided by the MS procedure, we apply the CG technique proposed by Cerrone et al. [8]. This previous study experimentally showed that compared to the solutions provided by a classic greedy algorithm, the solutions produced by CG contain more elements belonging at the same time to some optimal solution of the problem. Assuming that

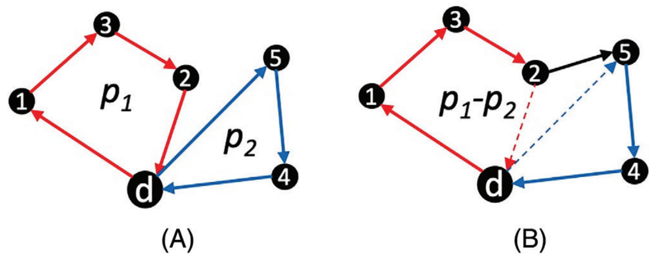


FIGURE 1 Routes p1 and p2 in (A) are merged into a single route shown in (B). Arcs (2, d) and (d, 5) are removed from the solution and arc (2,5) is inserted in their place. The cost difference between the two cases determines the savings obtained [Color figure can be viewed at wileyonlinelibrary.com]

CG will do the same for the GVRP currently under study, we apply the technique in conjunction with our randomized version of the savings algorithm.

The basic steps of the CG algorithm, applied to our MS procedure, can be summarized as follows:

Step 1: *A partial solution is built*.
Run MS algorithm;

(A) After the merge iteration, a pair of routes are merged according to their greedy function (the saving function). Each route defined in this merging operation is induced by a choice made by the algorithm; the chosen operations (choices) are stored in the order in which the MS algorithm performs them.
In our implementation, the sequence of merge operations is stored as an array of arcs *CL* for each arc used to merge two routes together.

(B) Drop some of the last β choices (usually $\beta = 3$).
Since CG works on incomplete solutions, we use the β parameter to define how many (of the latest route merging operations) have to be undone.

Step 2: *For a given number α of iterations (usually $\alpha = 1$ ride), the partial solution is modified by iteratively discarding the oldest merger of routes and making a new one.*
Drop the oldest merger of routes;
Choose a new one using the MS algorithm.
Since CG is used to give greedy algorithms the possibility of repeating the choices made several times, the α parameter is used to establish how many times to re-elaborate each choice. Each route merger operation performed by the MS is canceled and, if necessary, recreated exactly α times, in the same order in which the MS algorithm originally performed them.

Step 3: *The partial solution is completed*.
Run MS procedure initialized with the routes obtained after Step 2.

Figure 2 reports an example of the CG-MS algorithm execution. Figure 2.1–5 shows the execution of the MS algorithm, as described in step 1.A above. The algorithm starts with one route per customer and sequentially merges the routes by using arcs (1,3), (4,5), (3,2) and (2–4). In Figure 2.6 the execution of step 1.B is represented in the case of parameter $\beta = 2$. In Figure 2.7–10 the $\alpha|CL|$ iterations of step 2 are represented. In particular, Figure 2.7 shows the elimination of arc (1,3) from the solution and then the selection of arc (5,2) by using the greedy technique of MS; it should be noted that the total number of routes does not change in these iterations. Figure 2.8 and 9 shows, respectively, the elimination of arcs (4,5) and (5,2) from the solution and then the selection of arcs (2,3) and (3,1). Unlike in the previous iterations, Figure 2.10 shows us that, after arc (2,3) is removed from the solution, it can be selected again. Finally, Figure 2.11 and 12 are related to step 3; in this step the carousel scheme selects arcs to add to the solution until the stopping criterion is reached.

For each type of vehicle v the procedure described above is iterated 1000 times, and the frequency *FR* with which this vehicle is present in the solution generated is associated with each arc. Therefore, the value of *FR*, which is between 0 and 1000, represents an estimate that variable x_{ijv} is set to 1 in the final optimal solution.

Finally, we consider three different measures to select the most promising arcs belonging to graph G to be given as input to model (2)–(10). More precisely, for each node $i \in V$ we select only the first k arcs leaving from i having:

1) larger *FR* (*k/3 arcs are selected*);
2) smaller *cost/FR* (*k/3 new arcs are selected*);
3) larger *cost*FR* (*the remaining k/3 arcs are selected*).

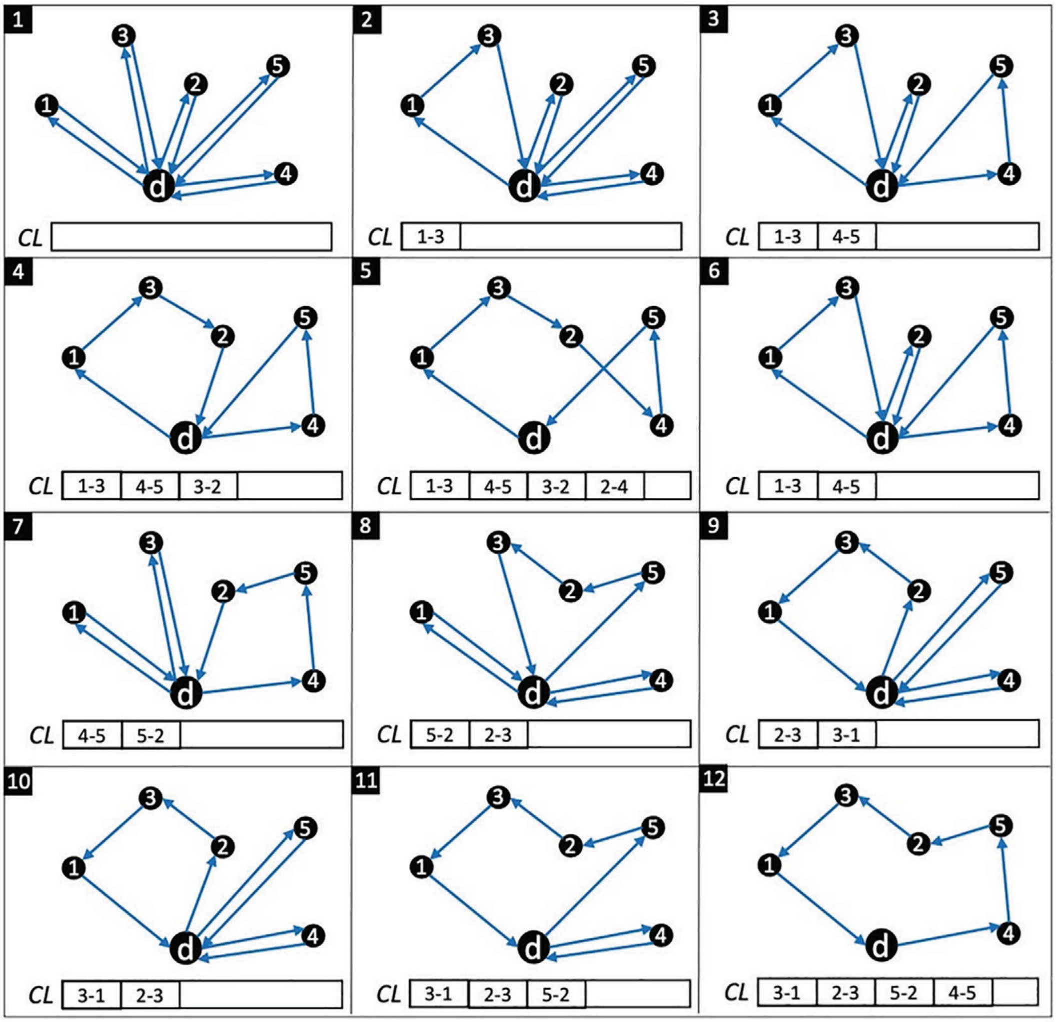


FIGURE 2 An example of the carousel greedy-max-saving algorithm execution [Color figure can be viewed at wileyonlinelibrary.com]

Note that (condition 1) implies that we select the arcs most frequently chosen by CG in its solutions. Condition 2 means that we choose the least expensive arcs with respect to the number of times they have been selected by CG. This policy obviously favors the arcs associated with smaller δ_{ijv} values. Finally, selecting arcs with a large *cost*FR* value means rewarding those arcs which, although expensive, have been selected often by CG.

In this procedure, the role of the parameter k is very important; in fact, as this value increases, the number of x_{ijv} and y_{ijv} variables present in the model will increase. This increases the probability of improving the value of the solution produced, but it greatly penalizes the execution time of the technique.

5 | TEST CASE: THE URBAN NETWORK OF THE CITY OF MILAN

In this section we report the computational experimentation of the proposed MILP model and matheuristic together with the evaluation of the impact of the penalty parameters δ_{ijv} for avoiding the crossing of bicycle lanes on the optimal routes.

We assume that set V of the fleet of nonhomogeneous vehicles is given by $V = V_1 \cup V_2 \cup V_3$. Table 1 reports the main characteristics of the three classes of vehicles considered in this study. In particular, the "Weight" and "Euro Class" columns give the information necessary to compute the coefficient c_{ijv} given in Equation (1) defining the traveling unit cost of each type of vehicle; values c_{ijv}, $\forall v \in V$ and $\forall v(i, j) \in A$ are then used in the objective function (2) of the proposed MILP model reported in Section 3. The "Size" and "Capacity (expressed in Boxes)" columns provide information about the loading capacity of the vehicles. In fact, for simplicity of representation, we consider that the goods to be delivered are contained in boxes of equal size and the boxes are such that they can also be transported by scooter; therefore, the capacity of each vehicle refers to the maximum number of boxes that vehicles can contain. Finally, the "Authorized to deliver" column expresses the compatibility of a vehicle of type v_t, to serve customer $i \in N$ with product p; this compatibility depends on the type of order, that is, fresh or dry products or a combination.

TABLE 2 Unit cost per km for each type of vehicle

Vehicle type	Traveling cost (€)	Air pollution cost (€)
1	0.020	—
2	0.142	0.039
3	0.246	0.053

FIGURE 3 (A) Private transportation network of Milan; (B) Map of the bicycle lanes [Color figure can be viewed at wileyonlinelibrary.com]

Table 2 reports the unit traveling cost t_v and the pollutant emission cost e_v of the considered types of vehicles. Note that all vehicles of the same type v have the same costs and capacity.

All the instances that are tested and reported here originated in the urban and metropolitan areas of the city of Milan (Italy). We chose Milan because it is one of the smartest European cities from a sustainable mobility point of view; at the same time, it is unfortunately also one of the cities most affected by the coronavirus pandemic. For this reason, the municipality is implementing tactical urban planning to offer alternatives to cars and to expand the cycle paths. In particular, the urban mobility plan foresees the construction of 35 km of new bike paths by spring 2021. Half of the cycle lanes will start from the suburbs of the city and will arrive near the center, thus offering a valid alternative to public transport. Note that cycling in Milan is easy because the city is not large (the distance between the districts located on the east and west sides is about 13 km) and it is flat. The street map of Milan is shown in Figure 3; in particular, Figure 3(A) depicts the roads traveled by private vehicles, while Figure 3(B) shows only the bicycle lanes. To better show the incidence of the bicycle lanes in the whole transportation network of the city, Figure 4 integrates these two maps into a single road map.

In order to generate the digraphs used as input for all instances of the problem, the GIS data related to the map of the city of Milan have been extracted from the repository OpenStreetMap [6, 18]. Using this data source, we created a road graph G containing 20.554 nodes and 36.073 arcs. For each arc $(i,j) \in A$ thus derived, some information relevant to the cycling conditions on the corresponding street have been additionally extracted. More precisely, a numerical value has been associated with each street in the most central area of Milan that can be covered by bicycle. This value, ranging from [1, 3], has been used to fix the street crossing penalty δ_{ijv}, according to the values reported in Table 3. In practice, since we decided to penalize commercial vehicle traffic on roads shared with bicycles, we have chosen to use an increasing value for the parameter δ_{ijv}, as the space dedicated to bicycles on the roadway decreases. We observe that no penalty is associated with vehicles of type $v = 1$, that is, electric scooters.

The streets of Milan mainly penalized by parameter $\delta_{ijv} = 3$ are enhanced in Figure 5 by red lines, while those penalized by parameter $\delta_{ijv} = 2$ are represented by orange lines; the more lightly penalized streets having parameter $\delta_{ijv} = 1$ are represented in green. A light color represents the streets without any active penalization. Figure 5 illustrates an instance of the problem: the depot, where the grocery goods are stored, is identified by a black dot inside a circle, while the green points are the residential customers (here $|C| = 50$).

FIGURE 4 Road map of the city of Milan [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 3 The chosen street crossing penalty values

	δ_{ij}		
Type of street	$v = 1$	$v = 2$	$v = 3$
Ample bicycle space	1	1.5	1.5
Sufficient bicycle space	1	2	2
Scarce bicycle space	1.5	3	3

It is worth noting that all of our computational experimentations are performed by randomly generating customer demand, splitting it among fresh, dry and mixed orders. Finally, in each instance customers are selected within the residential areas among nodes of the network. The number of customers varies from 20 to 140.

In the following, we report the computational results based on 11 different test instances, depending on the number of customers. All the code has been implemented in Java, using CPLEX version 12:8 as a solver. All tests were performed on a laptop with the following hardware features: MacBook Pro, processor Intel 2.3GHz, 4 core 32GB RAM.

In all instances assume that each type of vehicle v, $v = 1, 2, 3$, deliver to at most five routes; therefore, we set the corresponding values of m_v to $m_1 = 15$ and $m_2 = m_3 = 3$**,** which corresponds to the maximum allowed number of routes traveled by each type of vehicle equal to 45, 15, and 15, respectively. With regard to the demand, each customer $i \in C$ is randomly associated with a quantity $q_{ip} > 1$ such that the total number of boxes to deliver is equal to |C| *3, that is, three times the number of customers. This demand is then divided between 20% fresh and 80% dry products.

Table 4 reports the results of the proposed MILP model (2)–(10): the first columns report the results of model (2)–(10) when the street crossing penalty values δ_{ijv} are considered in the definition of the unit cost (see Equation (1)), while the last columns report the results when these penalty values are not considered. The column headings, which provide more detail, are as follows. “Customers” refers to the number of customers of the corresponding class of instance. For the information related to model (2)–(10), both with and without the penalty values, the column “Obj. value (2)” gives the optimal value of the objective function of the proposed MILP model (2)–(10), while “CPU Time (sec)” is the running time of the model required to find the optimal solution. Note that, since the maximum execution time has been fixed to 1 h, a value equal to 3600 s means that the model was not able to return the optimal solution within the given amount of time; in this case, for the last four instances we obtain optimality gaps equal to (0.49, 0.98, 3.25, 9.14)%, respectively. “Length (km)” is the sum of the length of all the generated routes, while “Routes” is the number of routes required to serve all customers. Finally, the “Vehicles ($v_1/v_2/v_3$)” column gives the value that corresponds, respectively, to the number of routes traveled by vehicles of type $v = 1, 2, 3$. We again note that the vehicles not suitable for delivering the type of products (fresh, dry or both fresh and dry) required by customers were not generated a priori and hence are not considered in the model.

Based on Table 4, it is easy to see that the retail company for which we are conducting the study would be able to guarantee the delivery of the required products to customers by maintaining its fleet of light commercial vehicles of type $v = 2$ and $v = 3$

FIGURE 5 An instance of the problem, showing penalization for each street [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 4 Computation results of the proposed MILP model

	MILP MODEL					MILP MODEL WITHOUT STREET CROSSING PENALTY				
Customers	Obj. value (2)	CPU time (sec)	Length (km)	Routes	Vehicles (1/2/3)	Obj. value (2)	CPU time (sec)	Length (km)	Routes	Vehicles (v1/v2/v3)
20	14.37	1	11.55	10	9/0/1	10.85	1	11.31	10	9/0/1
30	17.72	2	19.97	14	13/0/1	12.63	2	17.10	11	10/0/1
40	22.36	22	22.92	18	16/1/1	16.27	30	21.17	18	16/1/1
50	28.31	43	22.02	19	18/0/1	21.03	45	19.30	16	15/0/1
60	32.02	144	32.37	24	23/0/1	22.82	127	27.37	20	19/0/1
70	34.07	151	39.57	30	29/0/1	25.17	190	37.42	28	27/0/1
80	42.01	408	38.99	28	26/1/1	29.76	883	37.64	28	26/1/1
90	37.07	3600	40.30	30	28/0/2	28.39	3609	35.02	27	25/0/2
100	41.72	3608	48.02	42	40/0/2	30.41	3606	38.36	32	30/0/2
120	47.71	3600	37.45	29	25/2/2	33.51	3601	36.46	30	27/1/2
140	62.46	3600	67.27	55	52/1/2	44.62	3600	52.94	45	41/2/2

TABLE 5 Comparison (%) between the optimal solution obtained by the MILP model with and without the street crossing penalty with respect to the selected routes

	GAP		
Customers	Length (km)	Routes	$v = 1$
20	2%	0%	0%
30	17%	27%	30%
40	8%	0%	0%
50	14%	19%	20%
60	18%	20%	21%
70	6%	7%	7%
80	4%	0%	0%
AVG	10%	10%	11%

TABLE 6 Computation results of the proposed matheuristic with a time limit of 600 s

	Matheuristic						
Customers	*k*	Obj. Value (2)	% gap	CPU Time	Length	Routes	Vehicles
20	8	14.37	0.0%	1	11.55	10	9/0/1
30		17.72	0.0%	3	19.97	14	13/0/1
40		22.36	0.0%	20	22.99	18	16/1/1
50		28.39	0.3%	8	21.19	18	17/0/1
60		33.38	4.2%	56	28.41	19	17/1/1
70		36.51	7.2%	55	33.75	22	20/1/1
80		44.14	5.1%	514	34.20	21	18/2/1
90		35.33	4.5%	603	35.53	23	21/0/2
100		45.88	10.0%	604	32.59	22	20/0/2
120		45.04	1.3%	604	36.89	24	20/2/2
140		63.91	2.3%	607	40.00	24	19/3/2
20	10	14.37	0.0%	1	11.55	10	9/0/1
30		17.72	0.0%	3	19.97	14	13/0/1
40		22.36	0.0%	27	23.06	18	16/1/1
50		28.31	0.0%	18	22.02	19	18/0/1
60		32.45	1.3%	71	32.45	23	22/0/1
70		35.16	3.2%	73	38.20	26	24/1/1
80		43.48	3.5%	602	37.25	25	22/2/1
90		37.27	2.3%	330	37.86	26	23/1/2
100		45.66	9.4%	603	36.33	26	23/1/2
120		47.82	4.2%	604	38.64	26	23/1/2
140		64.94	4.0%	606	41.99	26	21/3/2
20	12	14.37	0.0%	1	11.55	10	9/0/1
30		17.72	0.0%	3	19.97	14	13/0/1
40		22.36	0.0%	21	22.94	18	16/1/1
50		28.31	0.0%	18	22.02	19	18/0/1
60		32.02	0.0%	61	32.38	24	23/0/1
70		34.12	0.1%	64	37.52	27	25/1/1
80		43.18	2.8%	526	39.75	27	24/2/1
90		36.61	0.5%	462	40.14	28	25/1/2
100		42.93	2.9%	605	40.85	32	29/1/2
120		43.94	0.6%	604	37.21	28	25/1/2
140		58.21	−6.8%	606	49.07	33	29/2/2
20	14	14.37	0.0%	1	11.55	10	9/0/1
30		17.72	0.0%	3	19.97	14	13/0/1
40		22.36	0.0%	31	22.92	18	16/1/1
50		28.31	0.0%	36	22.02	19	18/0/1
60		32.02	0.0%	79	32.37	24	23/0/1
70		34.11	0.1%	81	36.32	27	25/1/1
80		42.35	0.8%	167	39.46	28	26/1/1
90		36.48	0.2%	602	39.77	29	27/0/2
100		42.59	2.1%	603	42.48	34	32/0/2
120		42.13	−0.2%	605	38.81	28	25/1/2
140		59.64	−4.5%	607	48.05	33	29/2/2

and involving more scooters. Further, in order to avoid traveling along the bicycle paths as much as possible, the total length of the routes increases by 9.84% on average, with minimum and maximum values, respectively, of 2.13% and 18.26%. Note that these values refer only to the instances solved to optimality, that is, those with 20–80 customers. Moreover, when the penalty values δ_{ijv} are considered in the objective function (2), in the MILP model the overall distribution cost increases by about 8.5% on average, with the increase ranging from 2% to 15.5%. Further, considering that the average increase in the length of the route (9.84%) is greater than the increase in cost (8.52%), it can be assumed that the MILP model favors the use of less-expensive vehicles, such as electric scooters.

TABLE 7 Computation results of the proposed matheuristic with a time limit of 1800 s

	Matheuristic						
Customers	k	Obj. Value (2)	% gap	CPU Time	Length	Routes	Vehicles
20	8	14.37	0.0%	1	11.55	10	9/0/1
30		17.72	0.0%	2	19.97	14	13/0/1
40		22.36	0.0%	17	22.99	18	16/1/1
50		28.39	0.3%	6	21.19	18	17/0/1
60		33.38	4.2%	50	28.41	19	17/1/1
70		36.51	7.2%	45	33.75	22	20/1/1
80		44.14	5.1%	518	34.20	21	18/2/1
90		35.33	4.5%	1803	35.53	23	21/0/2
100		45.6	9.3%	1804	32.41	22	20/0/2
120		45.04	1.3%	1805	36.89	24	20/2/2
140		62.44	0.0%	1807	39.91	24	19/3/2
20	10	14.37	0.0%	1	11.55	10	9/0/1
30		17.72	0.0%	3	19.97	14	13/0/1
40		22.36	0.0%	25	23.06	18	16/1/1
50		28.31	0.0%	17	22.02	19	18/0/1
60		32.45	1.3%	61	32.45	23	22/0/1
70		35.16	3.2%	64	38.20	26	24/1/1
80		43.48	3.5%	887	37.25	25	22/2/1
90		37.27	2.3%	392	37.86	26	23/1/2
100		45.33	8.7%	1803	36.30	26	23/1/2
120		42.94	−1.4%	1806	36.48	26	23/1/2
140		60.85	−2.6%	1814	46.16	29	24/3/2
20	12	14.37	0.0%	1	11.55	10	9/0/1
30		17.72	0.0%	2	19.97	14	13/0/1
40		22.36	0.0%	23	22.94	18	16/1/1
50		28.31	0.0%	22	22.02	19	18/0/1
60		32.02	0.0%	72	32.38	24	23/0/1
70		34.12	0.1%	69	37.52	27	25/1/1
80		43.18	2.8%	649	39.75	27	24/2/1
90		36.61	0.5%	560	40.14	28	25/1/2
100		42.86	2.7%	1804	40.90	32	29/1/2
120		41.43	−1.3%	1807	36.20	28	25/1/2
140		57.98	−7.2%	1809	47.24	31	27/2/2
20	14	14.37	0.0%	1	11.55	10	9/0/1
30		17.72	0.0%	2	19.97	14	13/0/1
40		22.36	0.0%	30	22.92	18	16/1/1
50		28.31	0.0%	37	22.02	19	18/0/1
60		32.02	0.0%	83	32.37	24	23/0/1
70		34.11	0.1%	91	36.32	27	25/1/1
80		42.35	0.8%	196	39.46	28	26/1/1
90		36.48	0.2%	888	39.81	29	27/0/2
100		42.39	1.6%	1804	43.31	35	33/0/2
120		42.14	−0.3%	1806	37.11	28	25/1/2
140		58.54	−6.3%	1811	47.08	32	27/3/2

Table 5 provides a more detailed comparison between the results obtained by model (2)–(10) in two cases, one with and one without the penalty values δ_{ijv} for the transit of commercial vehicles near bicycle lanes. We see that when considering the penalty values δ_{ijv}, the overall length of the selected routes increases, on average, by 10%. This result is expected because in order to avoid traveling on the streets used by bicycles as much as possible, it is quite unavoidable that the route will become longer. The total number of routes increases as well. The reason for this increase can be seen from column "$v = 1$," which shows the number of routes traveled by type 1 vehicles when the penalty is active. In fact, these vehicles have $\delta_{ij1} = 1$, corresponding to no penalty, on almost all streets; thus the models favor their use, although these vehicles have a much smaller capacity than the

TABLE 8 MILP gap (in %) of the solution obtained by the matheuristic in 600 seconds of CPU time with respect to the MILP model

Customers	$k = 8$	$k = 10$	$k = 12$	$k = 14$	Average
20	0.0%	0.0%	0.0%	0.0%	0.0%
30	0.0%	0.0%	0.0%	0.0%	0.0%
40	0.0%	0.0%	0.0%	0.0%	0.0%
50	0.3%	0.0%	0.0%	0.0%	0.1%
60	4.2%	1.3%	0.0%	0.0%	1.4%
70	7.2%	3.2%	0.1%	0.1%	2.6%
80	5.1%	3.5%	2.8%	0.8%	3.0%
90	4.5%	2.3%	0.5%	0.2%	1.9%
100	10.0%	9.4%	2.9%	2.1%	6.1%
120	1.3%	4.2%	0.6%	−0.2%	1.5%
140	2.3%	4.0%	−6.8%	−4.5%	−1.3%
AVG	3.2%	2.5%	0.0%	−0.1%	

TABLE 9 MILP gap (in %) of the solution obtained by the matheuristic in 1800 seconds of CPU time with respect to the MILP model

Customers	$k = 8$	$k = 10$	$k = 12$	$k = 14$	Average
20	0.0%	0.0%	0.0%	0.0%	0.0%
30	0.0%	0.0%	0.0%	0.0%	0.0%
40	0.0%	0.0%	0.0%	0.0%	0.0%
50	0.3%	0.0%	0.0%	0.0%	0.1%
60	4.2%	1.3%	0.0%	0.0%	1.4%
70	7.2%	3.2%	0.1%	0.1%	2.6%
80	5.1%	3.5%	2.8%	0.8%	3.0%
90	4.5%	2.3%	0.5%	0.2%	1.9%
100	9.3%	8.7%	2.7%	1.6%	5.6%
120	1.3%	−1.4%	−1.3%	−0.3%	−0.4%
140	0.0%	−2.6%	−7.2%	−6.3%	−4.0%
AVG	2.9%	1.4%	−0.2%	−0.3%	

other types of vehicles. In particular, it is easy to observe that the increase in the number of routes corresponds to the increase of routes performed with type 1 vehicles. Note that Table 5 reports only the values up to 80 customers, for which we obtain the optimal solution by the MILP model within the time limit of 3600 s.

Tables 6 and 7 show the results of the proposed matheuristic with the processing time of the MILP model fixed at 10 and 30 min, respectively. In both tables we have hypothesized four values for the parameter k: $k = 8$, 10, 12, and 14. As in Table 4, for each scenario we report the value of the objective function (2), the optimality gap (in %) with respect to the MILP model (2)–(10) (after 3600 s), the CPU time (seconds), the total length traveled by the vehicles (km), and the number of routes and vehicles used for each type.

To better understand the data reported in Tables 6 and 7, Tables 8 and 9 provide a more in-depth analysis of the results, showing the values of the optimality gap as the number of customers and the value of the parameter k increase. A positive gap (enhanced in red) indicates that the matheuristic has produced a worse result than the MILP model, while a negative value (highlighted in green) indicates a better result than that of the MILP model. As expected, it is possible to have better values only in the case of instances that have not been resolved by the mathematical model within the time limit of 3600 s.

Table 8 shows that as k increases, the solutions produced improve significantly. In fact, as discussed in Section 4, as parameter k increases, the number of variables not fixed to zero by the matheuristic procedure increases. This means that the MILP model could potentially produce better solutions having more free variables. Thus, we begin to see a penalty for widening the research space by adding more variables. This behavior is evident when observing the scenario related to $k = 14$, where the improvement compared to the MILP model solutions obtained in 1 h of processing time is very low compared to the results obtained with the other values of k. Remembering that the results reported in Table 8 refer to the results reported in Table 4, we note that the optimal solution for the MILP model has only been obtained with instances up to 80 customers. To better verify this behavior, Table 9 shows the results obtained in 600 s of CPU time for the matheuristic procedure. In these cases, we are able to improve the results obtained for the largest instances and $k = 14$. As a further consideration, readers can note that as k increases, the

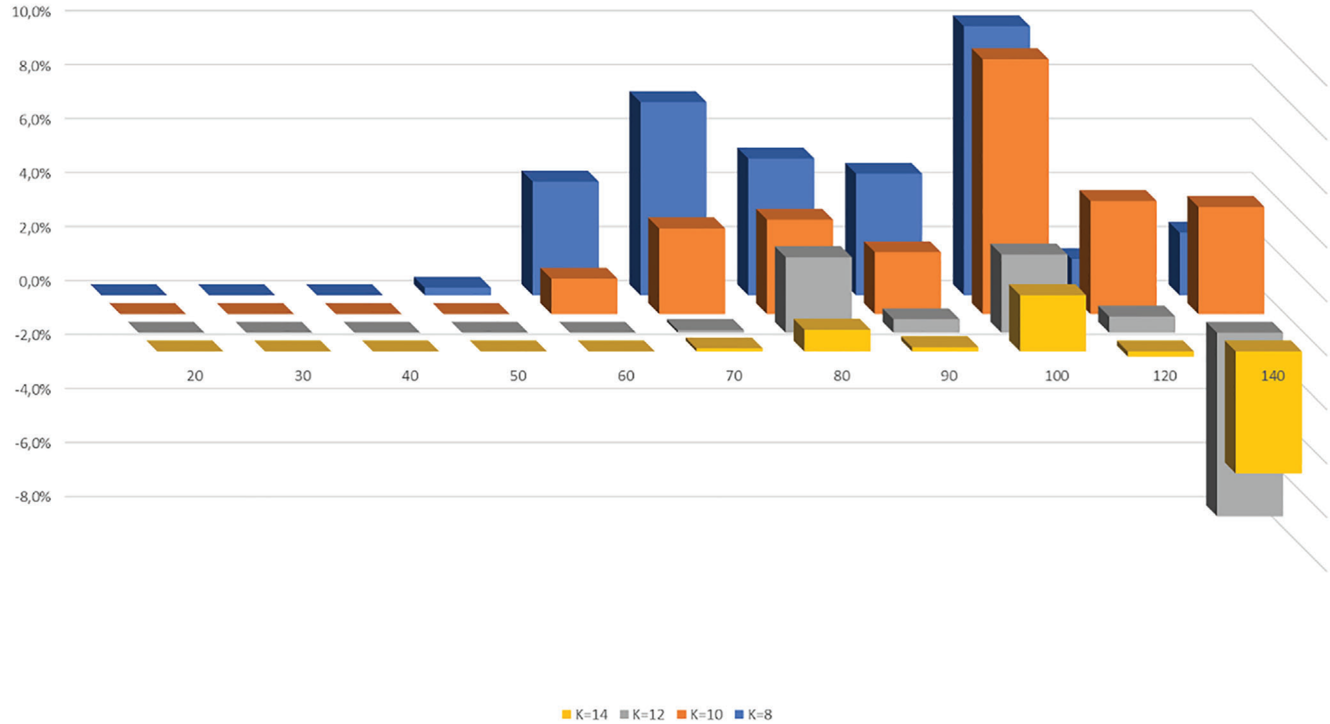


FIGURE 6 Optimality gap (in %) of the solution obtained by the matheuristic in 600 s of CPU time with respect to the MILP model [Color figure can be viewed at wileyonlinelibrary.com]

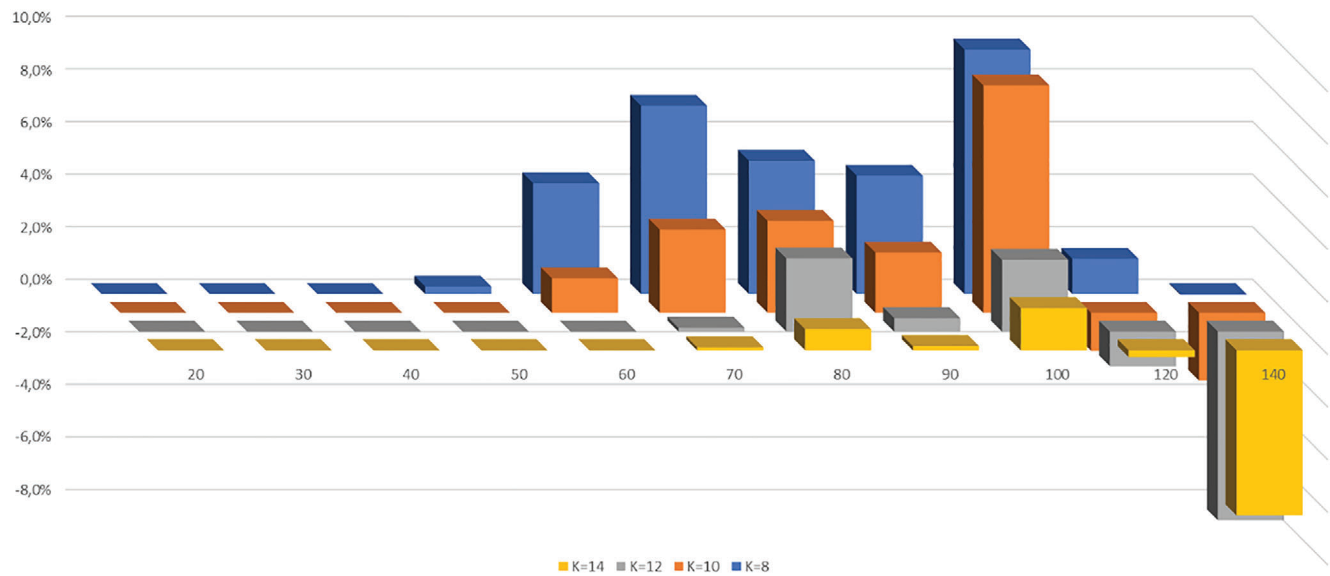


FIGURE 7 Optimality gap (in %) of the solution obtained by the matheuristic in 1800 seconds of CPU time with respect to the MILP model [Color figure can be viewed at wileyonlinelibrary.com]

processing time increases. Moreover, there are no significant differences between Tables 8 and 9 up to instances with more than 90 customers. This means that it is sufficient to set the maximum CPU time to 10 min. Finally, in the column corresponding to $k = 14$ up to 90 customers, the gap is always less than 1%.

At a first glance, it would seem obvious that the parameter k should be increased, and the execution time for the matheuristic should possibly be increased as well. However, looking at the results of the instance with 140 customers in Table 8 and the instances with 120 and 140 customers in Table 9, we observe that the best result is obtained with $k = 12$. From these tests it is clear that, once the time available for our procedure has been fixed, the parameter k must be chosen in relation to the size of the instance to solve. In particular and as we can see from the last two rows of Table 9, as the size of the instance increases, a smaller value k helps us to contain the growth in the number of variables and, consequently, to improve the quality of the solution produced in a given time limit.

The optimality gap of the proposed matheuristic with time limits of 600 and 1800 s, respectively, is represented in Figures 6 and 7. For all instances when the time limit is reached, we compared the matheuristic with the best solution produced by the MILP model in the corresponding time limit of 3600 s.

6 | CONCLUSION AND FUTURE WORK

In this paper we have addressed a variant of the GVRP of heterogeneous fleets, in which the objective function includes environmental impact cost components. In addition, we have introduced a new family of constraints that explicitly penalize routes that intersect or pass too close to bicycle lanes. For this problem we have proposed a mixed integer linear programming model and a matheuristic associated with it. We used this approach to analyze numerous instances related to the city of Milan (Italy). The present problem originated from the need for a B2C company to facilitate mobility on two wheels, as requested by the municipality of Milan to cope with the reduction in public transport capacity following the Covid-19 pandemic while also minimizing vehicular traffic congestion.

The reported results related to instances of different sizes, all derived from the graph representing the urban and metropolitan areas of Milan, which has more than 20 500 nodes and 36 000 arcs, demonstrate the effectiveness of the proposed matheuristic. Moreover, it is evident that the value of the penalty parameters, which depend on both the arcs of the graph and the type of vehicle chosen, affected the solutions obtained. As predicted, the need for safe cycling routes leads to a higher traveling cost for vehicle routing solutions. Yet surprisingly, the cost of the routes, considering the penalty coefficient, is on average only 9% greater than the cost of the routes obtained without considering the penalty, while the total length increase is only about 10% on average. In the future, we aim to carry out new computational experiments for instances related to different European cities. These new tests will allow us to verify whether the results obtained for the city of Milan are exceptional (due to the particular structure of the city road network) or whether they are independent of city-specific factors and it is therefore reasonable to expect similar results for other cities.

DATA AVAILABILITY STATEMENT

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

ORCID

Carmine Cerrone https://orcid.org/0000-0002-6243-4512

REFERENCES

[1] C. Archetti and M. G. Speranza, *A survey on matheuristics for routing problems*, EURO J. Comput. Optimiz. **2**(4) (2014), 223–246.

[2] C. Archetti, G. Guastaroba, and M. G. Speranza, *An ILP-refined tabu search for the directed profitable rural postman problem*, Discrete Appl. Math. **163** (2014), 3–16.

[3] M. Asghari and S. M. J. M. Al-e, *Green vehicle routing problem: A state-of-the-art review*, Int. J. Prod. Econ. **231** (2021), 107899.

[4] M. Behnke and T. Kirschstein, *The impact of path selection on ghg emissions in city logistics*, Transport. Res. Pt E. Logist. Transport. Rev. **106** (2017), 320–336.

[5] O. Bräysy, E. Martinez, Y. Nagata, and D. Soler, *The mixed capacitated general routing problem with turn penalties*, Expert Syst. Appl. **38** (2011), 12954–12966.

[6] G. Capobianco, C. Cerrone, R. Cerulli, and G. Felici, *Optimal Paths for Dual Propulsion Vehicles on Real Street Network Graphs*. In: Sforza A., Sterle C. (eds) *Optimization and Decision Science: Methodologies and Applications*. ODS 2017. Springer Proceedings in Mathematics & Statistics, vol 217. Springer, Cham. (2017) https://doi.org/10.1007/978-3-319-67308-0_52.

[7] F. Carrabs, C. Cerrone, R. Cerulli, and C. D'Ambrosio, *Improved Upper and Lower Bounds for the Close Enough Traveling Salesman Problem*. In: Au M., Castiglione A., Choo KK., Palmieri F., Li KC. (eds) *Green, Pervasive, and Cloud Computing*. GPC 2017. Lecture Notes in Computer Science, vol 10232. Springer, Cham. (2017) https://doi.org/10.1007/978-3-319-57186-7_14

[8] C. Cerrone, R. Cerulli, and B. Golden, *Carousel greedy: A generalized greedy algorithm with applications in optimization*, Comput. Oper. Res. **85** (2017), 97–112.

[9] R. Cerulli, R. P. Dameri, and A. Sciomachen, *Operations management in distribution networks within a smart city framework*, IMA J. Manag. Math. **29**(2) (2018), 189–205.

[10] G. Clarke and J. Wright, *Scheduling of vehicles for a central depot to a number of delivery points*, Oper. Res. **12**(4) (1964), 568–581.

[11] J. F. Ehmke, A. M. Campbell, and B. W. Thomas, *Optimizing for total costs in vehicle routing in urban areas*, Transp. Res. E **116** (2018), 242–265.

[12] S. Erdogan and E. Miller-Hooks, *A green vehicle routing problem*, Trans Res Part E **48** (2012), 100–114.

[13] I. A. Forma, T. Raviv, and M. Tzur, *A 3-step math heuristic for the static repositioning problem in bike-sharing systems*, Transp. Res. B Methodol. **71** (2015), 230–247.

[14] H. Kong, Q. Kang, W. Li, C. Liu, Y. Kang, and H. He, *A hybrid iterated carousel greedy algorithm for community detection in complex networks*, Phys. A Stat. Mech. Appl. **536** (2019), 122124.

[15] C. Lin, K. L. Choy, G. T. S. Ho, S. H. Chung, and H. Y. Lam, *Survey of green vehicle routing problem: Past and future trends*, Expert Syst. Appl. **41** (2014), 1118–1138.

[16] Maniezzo, V., Stützle, T. E Voß, S. (Eds.). *Matheuristics: Hybridizing metaheuristics and mathematical programming*. Springer, New York, 2009.
[17] D. Masłowskia, E. Kulińskab, and K. Kulińskac, *Application of routing methods in city logistics for sustainable road traffic*, Transport. Res. Proc. **39** (2019), 309–319.
[18] OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org. https://www. openstreetmap.org, 2017.
[19] H. Paessens, *The savings algorithm for the vehicle routing problems*, Eur. J. Oper. Res. **34**(3) (1988), 336–344.
[20] M. Yavuz, *An iterated beam search algorithm for the green vehicle routing problem*, Networks **69**(3) (2017), 317–328.
[21] S. Zhang, A. M. Campbell, and J. F. Ehmke, *Impact of congestion pricing schemes on costs and emissions of commercial fleets in urban areas*, Networks **73**(4) (2019), 466–489.

How to cite this article: C. Cerrone, R. Cerulli, and A. Sciomachen, *Grocery distribution plans in urban networks with street crossing penalties*, Networks. (2021), 1–16. https://doi.org/10.1002/net.22061