Biased-Randomized Iterated Local Search for a Multi-Period Vehicle Routing Problem with Price Discounts for Delivery Flexibility

Alejandro Estrada-Moreno[a], Martin Savelsbergh[b], Angel A. Juan[a,c], Javier Panadero[a,c]

[a]IN3 – Computer Science Department, Universitat Oberta de Catalunya, Barcelona, Spain
[b]H Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, USA
[c]Euncet Business School, Terrassa, Spain

Abstract

The multi-period vehicle routing problem (MPVRP) is an extension of the vehicle routing problem in which customer demands have to be delivered in one of several consecutive time periods, e.g., the days of a week. We introduce and explore a variant of the MPVRP in which the carrier offers a price discount in exchange for delivery flexibility. The carrier's goal is to minimize total costs, which consist of the distribution costs and the discounts paid. A biased-randomized iterated local search algorithm is proposed for its solution. The two-stage algorithm first quickly generates a number of promising customer-to-period assignments, and then intensively explores a subset of these assignments. An extensive computational study demonstrates the efficacy of the proposed algorithm and highlights the benefit of pricing for delivery flexibility in different settings.

Keywords: vehicle routing problem, multi-period, price discounts, biased-randomized heuristics, iterated local search

1. Introduction

The multi-period vehicle routing problem (MPVRP) (Francis et al., 2008) is an extension of the well-known vehicle routing problem (VRP)(Toth & Vigo, 2014; Caceres-Cruz et al., 2015) in which customer demands have to be delivered in one of several consecutive time periods, e.g., the days of a week. As observed in Archetti et al. (2015), in many practical settings there may exist some flexibility in the time of service and effectively exploiting that flexibility may result in significant cost savings. Therefore, we study a variant of the MPVRP in which a service provider offers a price discount in exchange for delivery flexibility. More specifically, we consider a setting in which customers place orders to be delivered during a planning horizon consisting of several consecutive days. Each customer specifies a demand quantity, a delivery location, and a *preferred* delivery day. The service provider offers a discount to its customers in

Email address: aestradamo@uoc.edu, martin.savelsbergh@isye.gatech.edu, ajuanp@uoc.edu, jpanaderom@uoc.edu (Javier Panadero)

exchange for flexibility in the timing of the delivery, e.g., the delivery is allowed to be made one day early or one day late, i.e., one day before or one day after the preferred day. This can be viewed as a form of collaboration because both parties benefit: a customer benefits directly by receiving a discount, and the service provider benefits indirectly by being able to exploit the additional flexibility to reduce distribution costs.

Settings in which delivery flexibility is provided in exchange for a price discount are becoming more common as such agreements are found in many provider-customer contracts. This may be due, in part, by the changing landscape of city logistics (Savelsbergh & Van Woensel, 2016). City councils encourage (or force) service providers to reduce the number of trips into the city, which may be achieved by serving customers in the same district/region in a single trip. Delivery flexibility is required for service providers to do so. While the "pricing for delivery flexibility" has been analyzed in the context of production planning (Li et al., 2012), there is a lack of similar studies in the VRP context. The primary goal of our research is to answer the following question: By how much can total costs be reduced for different levels of delivery flexibility and for different levels of price discounts?

We illustrate the potential benefits of delivery flexibility on a small example in Figure 1. An optimal delivery plan for three consecutive days, when deliveries have to take place of the preferred delivery day, is shown in Figure 1*a*. In order to simplify the visualization, we assume that the delivery routes on a particular day start and end at a different location. Figure 1*b* illustrates how delivering to some customers a day early or a day late can reduce the distribution costs.

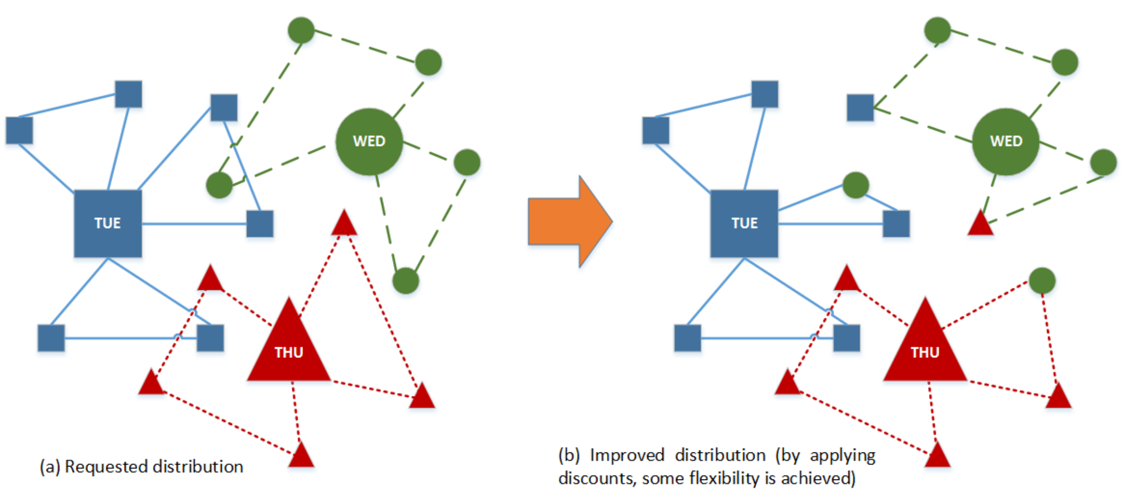

Figure 1: An example of the benefits of delivery flexibility.

To solve instances of this MPVRP variant, we propose a two-stage algorithm that combines iterated local search (ILS) (Lourenço et al., 2010) with biased-randomization (BR) techniques (Grasas et al., 2017). The algorithm will be shown to work well (i.e., producing near optimal solutions) in a variety of settings, from the most restricted setting, in which a customer can only be served on his preferred day, to the most flexible setting, in which a customer can be served on any day. As a consequence, we are able to establish, for an instance, the total costs as a function of customer flexibility (and for different price discounts) and, thus, provide valuable insights into the potential benefits of pricing for delivery flexibility.

To summarize, the main contributions of our research are threefold: *(i)* we introduce a new, but, in our view, relevant variant of the MPVRP, which allows the study of pricing for delivery flexibility; *(ii)* we propose a two-stage BR-ILS algorithm to solve instances of this MPVRP variant; and *(iii)* we analyze the relationship between delivery flexibility and distribution costs, extracting valuable managerial insights.

The remainder of this paper is organized as follows. Section 2 reviews relevant literature on related topics. Section 3 introduces the proposed solution approach. Section 4 presents the results of an extensive computational study and discusses insights obtained after an analysis of these results. Finally, Section 5 provides concluding remarks.

2. Literature Review

Early work on the MPVRP was carried out by Beltrami et al. (1974) considering two approaches to the problem: *(i)* developing routes and assigning them to delivery days; and *(ii)* assigning customers to delivery days and then routing each day independently. Russell & Igo (1979) proposed three approaches to the problem: *(i)* assigning customers to delivery days using a clustering algorithm –where the clusters for each day were formed around customers with a single allowable delivery combination; *(ii)* an adaptation of the single-day vehicle routing heuristic MTOUR, originally introduced in Russell (1977); and *(iii)* an adaptation of the single-day vehicle routing heuristic by Clarke & Wright (1964). In a similar way, Christofides & Beasley (1984) presented heuristic algorithms for the MPVRP based on an initial decision of delivery-day combinations for customers, followed by day-combination interchanges in an endeavor to reduce the total cost of the resulting VRPs. Hemmelmayr et al. (2009) proposed a new heuristic for the MPVRP based on variable neighborhood search. Their computational results show that the approach is competitive with other techniques. Likewise, Cacchiani et al. (2014) introduced a hybrid optimization algorithm, embedding both heuristic and exact components, for the MPVRP. The algorithm is based on the linear programming (LP) relaxation of a set-covering-like integer linear programming formulation of the problem. The whole solution method takes advantage of the LP-solution and applies techniques of fixing and releasing of the columns as a local search, making use of a tabu list to avoid cycling.

Several variants of the MPVRP are described in the literature. A classification of the different versions can be found in a survey by Mourgaya & Vanderbeck (2007). The authors distinguished between different objective functions. Typical goals that are considered are: minimization of the total traveled distance, minimization of accumulated driving time, or minimization of the total transportation cost. However, many other features can be part of the objective function as well, e.g.: regionalization of routes, workload balance among different vehicles, number of required vehicles, service quality, etc. Likewise, constraints can also be classified in three categories: *(i)* the ones related to the planning of the visits (different frequencies, restrictions on certain days, etc.); *(ii)* those related to the type of demand (deterministic or stochastic); and *(iii)* the ones associated with the vehicles (e.g.: capacity constraints, fleet composition, etc.).

Next, some extensions of the basic MPVRP are presented. Angelelli & Speranza (2002) addressed a generalization of the MPVRP in which some intermediate facilities exist and where vehicles renew entirely their capacity and return to the depot only when the work shift is over. In this work, a tabu search (TS) algorithm is proposed for solving the problem with the aim of minimizing the total length of the routes traveled by the vehicles. Francis et al. (2006) and Francis et al. (2007) analyze the MPVRP with service choice, in which the service frequency is a model decision variable. In the former article, the authors introduce a mathematical model and develop an exact solution method as well as a heuristic solving procedure. In the latter article, they propose a TS algorithm to study the trade-offs between operational flexibility (in terms of costs and customer service benefits) and complexity (in terms of modeling and implementation difficulty) for this periodic VRP. Zäpfel & Bögl (2008) considered a short-range weekly planning by postal companies that must decide about pickup tours and delivery tour for fluctuating volume (number of shipments). The problem also considered time windows for the demand points, variable vehicle capacities and personnel planning, as well as outsourcing decision for tours and drivers. A hybrid metaheuristic was proposed. Similarly, Alonso et al. (2008) studied a VRP including the possibility that a vehicle can do more than one route per day –as long as the maximum daily operation time is not exceeded. In addition, some constraints related to the accessibility of the vehicles to the customers are considered too. A TS algorithm was also used in this case.

Other relevant variants are those from Wen et al. (2010) and Albareda-Sambola et al. (2014), which are based on the concept of dynamic orders. The former considered the dynamic MPVRP, in which customers place orders dynamically over a planning horizon consisting of several periods. The distributor must plan its delivery routes over several days in order to minimize the routing cost and customer waiting, as well as to balance the daily workload over the planning horizon. The latter introduced the dynamic MPVRP with probabilistic information, in which calls for service arrive throughout a discrete time horizon and must be fulfilled within a time window that comprises several time periods. In addition to known information about customers (request time windows and distances), it is assumed that probabilistic information regarding calls for service in future time periods is available. Thus, at each time period two decisions must be made: *(i)* the choice of the subset of pending service requests to satisfy; and *(ii)* the design of the vehicle routes.

In Pacheco et al. (2012), the authors propose a mixed-integer linear model for a MPVRP based on a real-life case. Also, a GRASP with path-relinking metaheuristic is proposed to solve the problem. Using real-life data, they show that the new approach is able to significantly lower the enterprise distribution costs. Rahimi-Vahed et al. (2013) considered a variant of the MPVRP with several depots and proposed an efficient path relinking algorithm to tackle it. This algorithm is able to address the problem in two different settings: *(i)* as a stand-alone algorithm; and *(ii)* as part of a co-operative search algorithm called integrative co-operative search. Zhang et al. (2013) defined a new extension of the MPVRP with profit, and designed an effective memetic algorithm with a giant-tour representation to solve it. The goal is to determine routes for a set of vehicles that maximizes profitability from visited locations, based on the conditions that vehicles can only travel during stipulated working hours within each period. They also considered that the vehicles are only required to return to the depot at the end of the last period. Likewise, Vidal et al. (2012) proposed an algorithmic framework that successfully addressed three vehicle routing problems: the multi-depot VRP, the multi-period VRP, and the multi-depot periodic VRP with constrained route duration.

Recently, Luo et al. (2015) investigated a new variant called the MPVRP with time windows and limited visiting quota, which requires that any customer can be served by at most a certain number of different vehicles over the planning horizon. Another variant is the MPVRP with due dates, where customers have to be served between a release date and a due date (Archetti et al., 2015). In this problem, customers with due dates exceeding the planning period may be postponed at a cost and a fleet of capacitated vehicles is available to perform the distribution in each day of the planning period. The aim of this problem is to find vehicle routes for each day such that the total cost of the distribution is minimized. In some sense, our paper extends the work of Archetti et al. (2015) by introducing the possibility of applying incentives to the customers in order to gain more flexibility when designing the delivery routing plans. Finally, Archetti et al. (2017) introduce the flexible MPVRP, in which a distribution plan has to be designed over a time horizon. These authors include some flexibility in the service frequencies and schedules, and also establish a link between this flexible MPVRP and the inventory routing problem. The authors propose a mathematical formulation and use Cplex to solve some small-scale instances, but they also recognize that larger instances require the use of heuristic-based approaches. Precisely, our paper proposes a two-stage metaheuristic algorithm to deal with the MPVRP with flexibility in the delivery dates in turn of price discounts. However, we do not link our MPVRP version to the inventory routing problem but, instead, we create similarities with the multi-depot VRP.

3. A Two-Stage BR-ILS Algorithm

To solve the MPVRP with price discounts for delivery flexibility, a two-stage metaheuristic approach has been developed (Figure 2). The two-stage schema has been adopted since this is a bi-level problem, where the initial assignment of customers to days can have a strong impact on the ulterior routing plans. The solution approach combines BR techniques with an ILS framework (which includes a perturbation stage and a local search stage), hence will be referred to as BR-ILS. In the first-stage, a series of “promising” customer-to-day *assignment maps* (i.e., assignment plans) are generated, and their associated delivery costs are estimated using a fast routing heuristic. In the second stage, these delivery costs are refined, i.e., computed with a higher accuracy, for the most promising customer-to-day assignment maps via a more intensive routing procedure. Both stages are explained next in more detail.

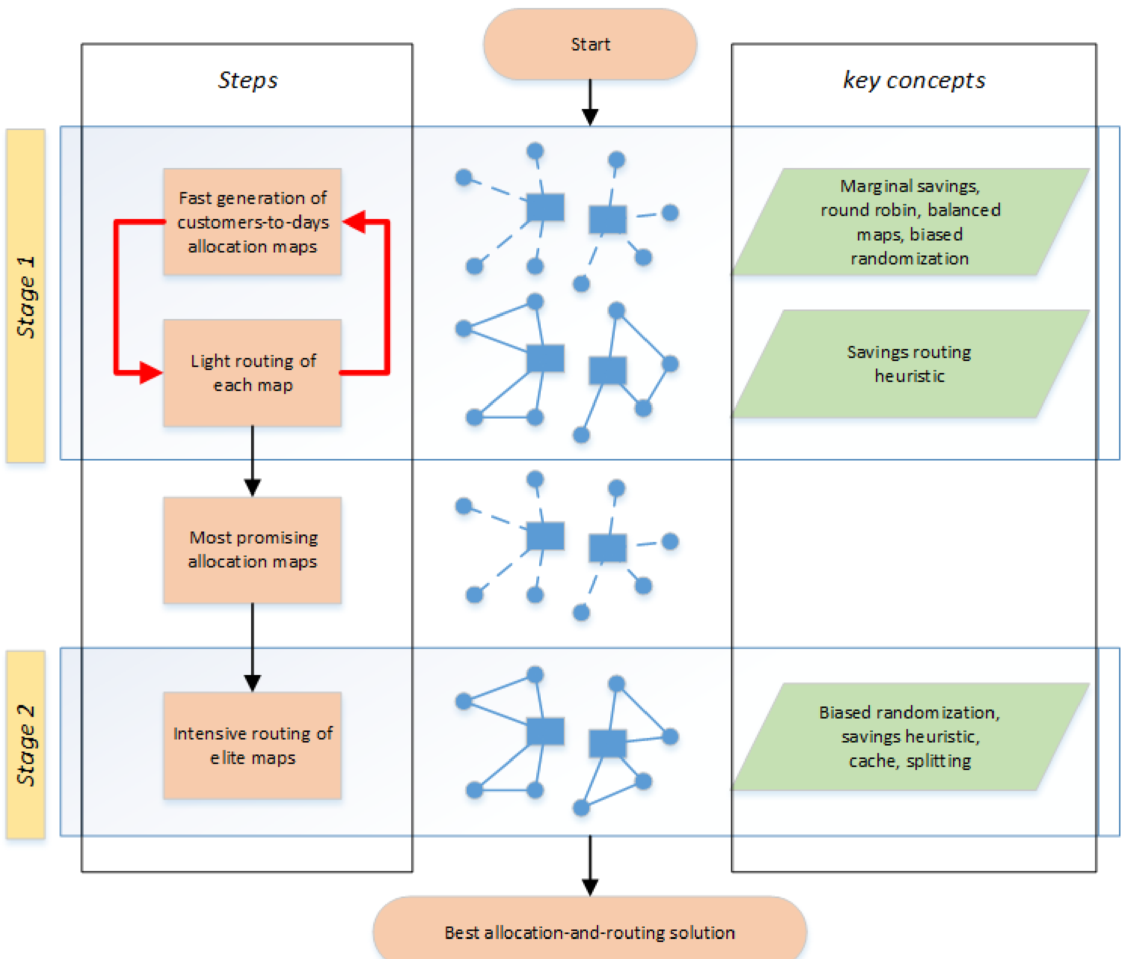

Figure 2: A two-stage solution approach.

3.1. Generating Promising Customer-to-Day Assignment Maps

The first stage of our algorithm consists of: *(i)* fast generation of many promising customer-to-day assignment maps, i.e., maps that, by construction, are good candidates to have low delivery costs – considering both routing costs and "penalty" costs, representing any discounts paid; *(ii)* estimation of the routing cost for each of the assignment maps; and *(iii)* sorting the maps in order of nondecreasing estimated total costs, so that a small set of "elite" maps is selected for further processing (in the second stage). Specifics are provided next.

- To be able to generate many promising customer-to-day assignment maps, we start by computing s_{id}, the "desirability" of delivering to customer i on day d (for each customer i and each feasible delivery day d). Three schemes have been developed to determine this desirability index (two of them are static in the way they are computed, while the third one is dynamically computed):

  1. *k-nearest:* In a high-quality solution, a customer i is likely to be visited by a vehicle that also visits customers near i. Therefore, we approximate the cost c_{id} of delivering to customer i on day d as the sum of the distances from customer i to the k nearest customers having d as their preferred delivery day, since these customers are more likely to be visited on day d. Reasonable choices for k are 2 and 3 (in our experiments, we have used $k = 3$). We now define the desirability s_{id} of visiting customer i on day d to be $c_{id^*} - c_{id}$, where $d^* = \arg\min_{\bar{d} \neq d} c_{i\bar{d}}$, i.e., the best alternative day.
  2. *depot-based:* Alternatively, the cost c_{id} of delivering to customer i on day d is approximated by the distance from customer i to the depot. Again, the desirability s_{id} of visiting customer i on day d is defined to be $c_{id^*} - c_{id}$, where $d^* = \arg\min_{\bar{d} \neq d} c_{i\bar{d}}$.
  3. *dynamic-nearest:* Finally, another option is to approximate the cost c_{id} by the distance of customer i to the nearest customer j that has already been assigned to day d for delivery (also accounting for the discount fee). Note that this cost approximation is dynamic and has to be updated each time a new customer-to-day assignment is made.

- Next, for each day, a list of potential customers to visit in nonincreasing order of desirability is created (the list contains only those customers who can be visited on the corresponding day). Then, a round-robin selection process is started in which days "choose", in sequential turns, the customers to be assigned to them. When implemented in code, this is a heuristic procedure that can be run in milliseconds using a standard computer. However, since it is also a deterministic procedure, it generates exactly the same map every time it is run. To overcome this limitation, biased-randomization techniques are used (Juan et al., 2013), i.e., instead of always selecting the candidate at the top of the list, a skewed probability distribution is employed to introduce some randomness in the process without destroying the logic behind the sorting criterion. In our case, the Geometric probability distribution is utilized, since it has been well-tested in different vehicle routing and scheduling applications (Juan et al., 2011, 2014; Dominguez et al., 2016a,b).

- When assigning customers to days, it might be convenient to seek a balanced distribution of the demand served across these days. This is especially important for instances in which the total demand to be served during the planning period is close to the total delivery capacity. For these "tight" instances, an unbalanced distribution of the demand across days might cause the delivery plan to become unfeasible.

Note that it is easy to generate balanced assignment maps by adapting the previously described round-robin process as follows: instead of employing a pure round robin process, or even a random one, balanced maps are generated by granting the selection turn to the day with the highest available capacity.

- Since it is critical to evaluate as many customer-to-day assignment maps as possible in the available computing time, every time a map is generated its routing cost is estimated by using a fast routing heuristic – the well-known savings procedure in our case (Clarke & Wright, 1964). Even though the routing cost obtained is only an estimate of the real routing cost associated with a map, it allows us to compare maps and then run a more intensive routing algorithm over an elite subset of the maps in the second stage.

Algorithm 1 provides a more detailed description on how these different components are combined in a BR-ILS framework. Apart from the inputs, a reduced number of parameters are used in this algorithm (the values of these parameters have been chosen after a quick trial-and-error fine-tuning process):

- $maxIter$: maximum number of iterations allowed; 300,000 in our tests.
- p: percentage of the $baseMap$ destroyed in the perturbation phase; in our computational study, at each iteration p is randomly chosen in the interval (30, 70).
- α: parameter of the Geometric distribution used in the BR round-robin process; in our experiments, at each iteration α is randomly chosen in the interval (0.5, 0.8).
- n: number of $eliteMaps$ to be selected from $promisingMaps$; 10 in our computational study.
- β: parameter of the Geometric distribution utilized in the BR routing algorithm; in our tests, β is randomly chosen within the interval (0.10, 0.20).

3.2. Refining the Routing Plans for Elite Assignment Maps

For each of the elite maps obtained in the first stage, a more intensive routing algorithm is applied. Of course, the number of elite maps to consider will depend on the available time and computing resources – note that the routing of these elite maps is a task that can be run in parallel, using different threads, cores, or computers. The vehicle routing algorithm employed in our tests is the SR-GCWS-CS one proposed in Juan et al. (2011). It encapsulates a biased-randomized version of the popular savings heuristic into a multi-start process, which is noticeably enhanced by the use of a cache memory and a 'splitting' technique. The cache procedure relies on a hash map data structure that keeps the best-found path to connect a given set of customers. It also contains a fast local search that is similar to a 2-opt one. The splitting technique

Algorithm 1 A two-stage BR-ILS algorithm for the MPVRP with delivery flexibility.

```
function BR-ILS(customers, days, shift, discount, maxIter, p, α, n, β)
% Stage 1: generation of promising customer-to-day assignment maps.
    for each (d ∈ days) do
        for each (i ∈ customers) do
            s_id ← calcDesirability(d, i, shift, discount)
        end for
        priorityLists(d) ← sortCustomersByDesirability(d)
    end for
    initialMap ← roundRobin(days, customers, priorityLists, α)
    estimatedCost(initialMap) ← routingHeuristic(initialMap)
    baseMap ← initialMap
    promisingMaps ← add(initialMap)
    iter ← 0
    while (iter < maxIter) do
        newMap ← perturbate(baseMap, p, α) % BR round-robin process.
        newMap ← localSearch(newMap) % a cache with inner-loops operator.
        cost(newMap) ← lightRoutingHeuristic(newMap)
        delta ← cost(baseMap) − cost(newMap)
        if (delta ≥ 0) then % newMap improves baseMap
            promisingMaps ← add(0, newMap)
            baseMap ← newMap, credit ← delta
        else
            if (−delta ≤ credit) then % acceptance criterion.
                baseMap ← newMap, credit ← 0
            end if
        end if
        iter ← iter + 1
    end while
% Stage 2: refinement of routing plans for elite allocation maps.
    eliteMaps ← selectBestMaps(promisingMaps, n)
    cost(bestMap) ← ∞, bestMap ← ∅
    for {map in eliteMaps} do
        cost(map) ← intensiveRoutingAlgorithm(map, β) % BR algorithm with cache and splitting.
        if (cost(map) < cost(bestMap)) then
            cost(bestMap) ← cost(map), bestMap ← map
        end if
    end for
    return bestMap
end function
```

can be seen as a divide-and-conquer procedure: each complete solution generated by the constructive process is divided into several disjoint sub-solutions using a proximity criterion; then, each of these sub-solutions are recomputed as if they were a new VRP of smaller size. If a better solution is found for any of the subproblems, then the global solution will be improved. The SR-GCWS-CS algorithm has been enriched with an additional local search process, by keeping, for each customer and for each day a delivery to the customer can be made, a list with the l nearest neighbors. This allows for fast evaluation of reassigning customers to different days. In our computational study, we have set $l = 2 \cdot nCust/(nDays \cdot nVeh)$, where $nCust$, $nDays$, and $nVeh$ are the number of customers, days, and vehicles, respectively.

4. Computational Experiments

The methodology described in this paper has been implemented as a Java application. All the experiments have been run in an Intel Xeon E5-2630 v4 CPU at 2.20 GHz and 32 GB RAM.

Being a new MPVRP variant, with price discounts in exchange for delivery flexibility, no benchmark instances exists. Therefore, we decided to adapt a well-known set of benchmark instances from the multi-depot vehicle routing problem (MDVRP). Apart from giving us a base dataset with highly-competitive best-known solutions, the use of this MDVRP benchmark is not unreasonable as the MDVRP has certain similarities with the MPVRP with price discounts for delivery flexibility. More specifically, when we consider each depot as representing a different day of the week (Monday, Tuesday, etc.), and we assume that no discounts have to be offered to customers to deviate from their preferred delivery day (i.e., we are free to choose the delivery day), then the only difference is the location of the depots, which are geographically dispersed in the MDVRP and geographically co-located in the MPVRP with price discounts for delivery flexibility.

Therefore, we have used the benchmark MDVRP instances described in Cordeau et al. (1997) (instances $p01$ to $p23$) and the benchmark MDVRP instances introduced in Pisinger & Ropke (2007) (instances $pr01$ to $pr10$). These instances assume a limited fleet of available vehicles per day, and, in some cases, a maximum route length. It is quite easy to modify these MDVRP instances so that they better reflect the setting of interest, in which the depots are geographically co-located. Specifically, we replace the depots in each MDVRP instance by a single depot located at the geometric center of the original ones (see Figure 3). Note that the 33 new instances retain most of the characteristics of the original ones (fleet and depot capacities, customers locations, etc.). However, any maximum route length constraints have been omitted to avoid any infeasibility (due to changes in travel distances resulting from re-locating the depots). For each instance, the preferred delivery day for a customer is fixed in advanced at random (once fixed, the preferred day cannot be modified during the experiments, so if a customer is delivered any other day then a discount has to be applied).

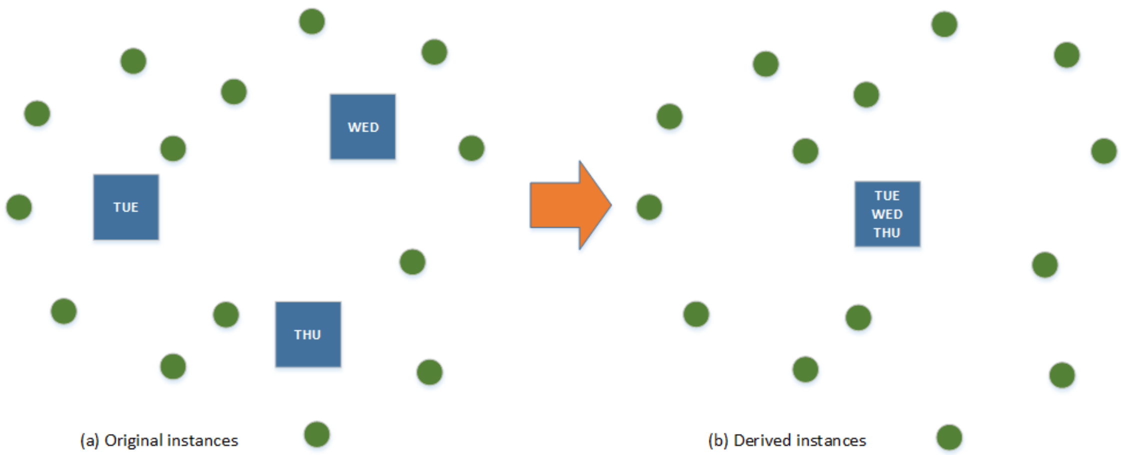

Figure 3: Generating MPVRP-PD instances from MDVRP instances.

4.1. Experiment 1: Instances with geographically dispersed (non-overlapping) depots

To assess the impact of offering price discounts for delivery flexibility, the following scenarios have been considered:

- Best-case or total flexibility scenario: Here, it is assumed that customers do not have any preferences regarding the delivery day (i.e., the preferred delivery day is ignored). Therefore, it is possible to assign a customer to any day without having to pay a price discount. Note that in this "ideal" (and probably unrealistic) scenario, we are solving an MDVRP.

  Note that by considering this best-case scenario (for instances with geographically dispersed depots), we are able to assess the performance of our BR-ILS algorithm as near-optimal solutions are known and reported in the MDVRP literature.

- Worse-case or zero flexibility scenario: Here, it is assumed that customers are unwilling to deviate from their preferred delivery day, no matter what price discounts are offered, and it is not possible to assign them to another day. Note that this lack of flexibility will generate more costly delivery plans – compared to the previous scenario – and could even cause some instances to become infeasible (depending on available delivery capacity). Also, since customers have to be served on a given day, a solution can be obtained by solving a number of VRPs (one for each day). Therefore, the solution we report for this worst-case scenario is obtained by solving the VRPs using the SR-GCWS-CS algorithm proposed in Juan et al. (2011).

- Intermediate or partially flexible scenarios: Finally, we consider different intermediate scenarios with different degrees of flexibility (e.g., different levels of discount fees combined with the option to deliver 1 or 2 days early or late relative to the preferred delivery day of customers). For higher degrees of flexibility, we should obtain solutions of increasing quality – higher than the quality of the solutions obtained for the worse-case scenario but, at the same time, lower than the quality of the solutions obtained for the ideal scenario with total flexibility. By comparing these intermediate scenarios with the extreme ones, we expect to answer the following managerial question: how much can the delivery plans be improved for different degrees of flexibility? In our experiments, we assume a price of $p = 5$ monetary units (m.u.) for each unit of product. If a customer i, initially assigned to a day d, is reallocated to $d-1$ or to $d+1$, then she receives a discount on the unit price p. In order to consider different scenarios, the level of discount is a design parameter with three different levels: 0.1 m.u. per unit of product (low-penalty level), 0.5 m.u. per unit of product (medium-penalty level) and 1.0 m.u. per unit of product (high-penalty level). Also, if customer i is reassigned to day $d-2$ or $d+2$ then the customer receives the 1-day discount plus an extra 25% of this 1-day discount.

The results of these tests are summarized in Figure 4, which shows a multiple-boxplot that allows to visually compare the costs that can be achieved in the different scenarios analyzed in the non-overlapping (NO) case. Here, NO-F refers to the non-overlapping case with full flexibility (F), while NO-2-H refers to the non-overlapping case with a 2-day shift (2) and high (H) flexibility. Notice that our best solution (OBS) for the full-flexibility scenario (best-case) provides an average gap which is below 0.3% with respect to the best-known solution (BKS) for the MDVRP. This highlights our algorithm's ability to produce high-quality solutions. On the other extreme, OBS for the zero-flexibility scenario provides a gap over 12% with respect to the BKS of the ideal scenario. Then, as we progressively consider higher flexibility levels, the associated OBS values show lower average gaps: about 9% in the case of 1-day and high discount fees, about 8% in the case of 2-day and high discount fees, about 8% in the case of 1-day and medium discount fees, etc.

4.2. Experiment 2: Instances with geographically co-located (overlapping) depots

The previous experiment allowed us to compare the results provided by our algorithm with two benchmarks that acted as extreme scenarios of the MPVRP: the MDVRP (full-flexibility scenario) and a series of independent VRPs (zero-flexibility scenario). The small gap found between the solutions provided by our algorithm for the full-flexibility scenario and the BKS reported for the MDVRP instances validates its performance. However, in order to use the MDVRP benchmark we needed to assume geographically dispersed depots. In this subsection, we focus on the setting of interest, the MPVRP with price discounts for delivery flexibility, in which the depot is located in the same location in each period.

Tables 1 and 2 show the results obtained for this co-located (overlapping depots) scenario when 1-day and 2-day demand shifts are allowed, respectively. The following descriptive data is provided: instance name and number of days D (where each day corresponds to a different depot); next, our best solution (OBS) for the best-case scenario (with full flexibility) are shown.; afterwards, the table includes our best solutions for

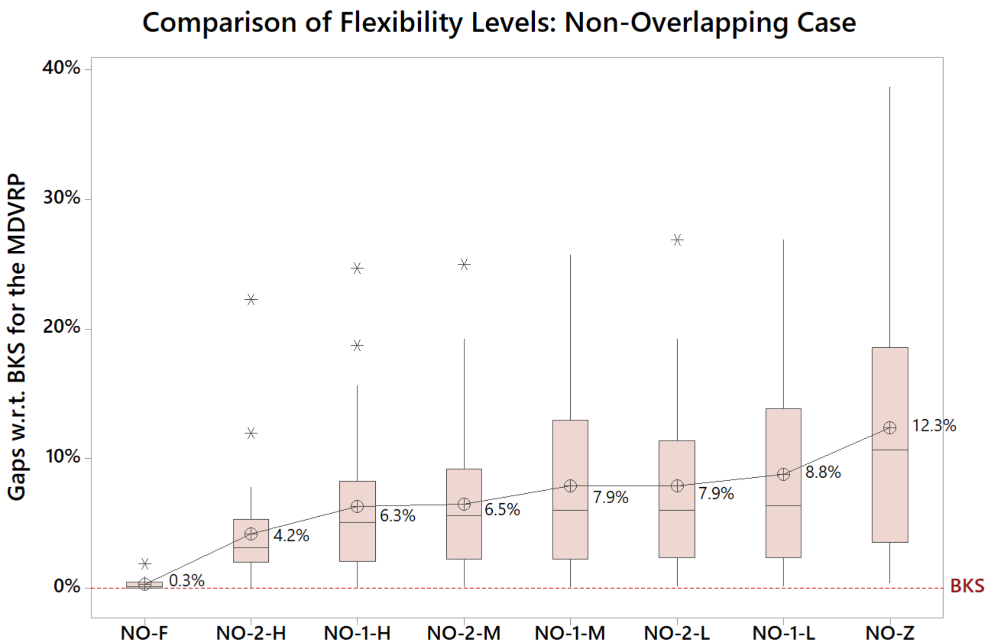

Figure 4: A visual comparison of the different scenarios assuming geographically dispersed depots.

the different intermediate scenarios: *(i)* high flexibility, with one shifting day and a low discount rate (low penalty level); *(ii)* medium flexibility, with two shifting days and a medium discount rate (medium penalty level); and *(iii)* low flexibility, with one shifting day and a high discount rate (high penalty level); OBS values refer to the distribution cost, and they include the 'penalty' cost that the company incurs when applying discount fees to customers in exchange for reallocation of the preferred distribution day; finally, our best solutions for the worse-case scenario with zero flexibility are presented. As previously discussed, these last solutions correspond to the computation of a series of VRPs, and since we are using a well-tested routing algorithm, the results are expected to be near-optimal. As expected, the results show how by increasing the flexibility level the OBS values always move away from the worse-case scenario and approach the best-case one.

For the overlapping case, Figure 5 plots the 95% confidence intervals associated with the average gaps between the *depot-based* (static) and the *dynamic-nearest* schemes for the desirability concept previously introduced. Here, O-1-H refers to the overlapping case (O) with a 1-day shifting and a high (H) flexibility level (i.e., one with a low penalty cost). Most of the improvements are significant for a standard confidence level, although they are relatively small (between 0.1% and 0.2% in all cases).

For a 1-day and 2-day shifts, respectively, Tables 3 and 4 show a comparison between the three schemes employed to determine the desirability concept during the first stage of our algorithm, the *depot-based* (dB), *dynamic-nearest* (dN), and *k-nearest* (kN). Notice that the results are very similar regardless the strategy

Table 1: Results obtained for the overlapping case and 1-day shift.

		Best-case Scenario with 100% flexibility	Intermediate Scenario with high flexibility (days = 1; penalty = Low)			Intermediate Scenario with medium flexibility (days = 1; penalty = Medium)			Intermediate Scenario with low flexibility (days = 1; penalty = High)			Worse-case Scenario with 0% flexibility	
Inst.	Ndays	OBS (1)	OBS COST (2)	Discount	GAP (2)-(1)	OBS COST (3)	Discount	GAP (3)-(1)	OBS COST (4)	Discount	GAP (4)-(1)	OBS COST (5)	GAP (5)-(1)
p01	4	746.64	805.53	0.96%	7.89%	834.43	3.72%	11.76%	840.02	0.00%	12.51%	840.02	12.51%
p02	4	529.72	577.97	2.70%	9.11%	624.89	4.00%	17.97%	641.37	0.00%	21.08%	641.37	21.08%
p03	5	850.46	896.51	2.00%	5.41%	916.53	0.55%	7.77%	921.53	1.09%	8.36%	922.39	8.46%
p04	2	1,171.61	1,175.71	0.35%	0.35%	1,191.87	0.00%	1.73%	1,191.87	0.00%	1.73%	1,191.87	1.73%
p05	2	830.05	834.25	0.50%	0.51%	836.94	0.18%	0.83%	838.44	0.36%	1.01%	838.96	1.07%
p06	3	1,171.36	1,172.56	0.10%	0.10%	1,177.36	0.51%	0.51%	1,183.36	1.01%	1.02%	1,190.42	1.63%
p07	4	1,184.42	1,213.55	0.63%	2.46%	1,242.09	0.40%	4.87%	1,247.09	0.80%	5.29%	1,264.13	6.73%
p08	2	5,307.75	5,332.73	0.31%	0.47%	5,358.73	0.17%	0.96%	5,364.12	0.34%	1.06%	5,381.24	1.38%
p09	3	4,515.76	4,721.35	1.51%	4.55%	4,823.23	1.50%	6.81%	4,864.68	0.00%	7.73%	4,864.68	7.73%
p10	4	5,281.13	5,477.54	0.35%	3.72%	5,516.89	0.22%	4.46%	5,532.13	0.00%	4.75%	5,532.13	4.75%
p11	5	5,367.76	5,447.78	1.24%	1.49%	5,531.91	0.28%	3.06%	5,547.41	0.56%	3.35%	5,625.84	4.81%
p12	2	1,789.01	1,804.37	0.21%	0.86%	1,819.00	0.99%	1.68%	1,835.91	1.96%	2.62%	1,836.53	2.66%
p13	2	1,789.01	1,804.37	0.21%	0.86%	1,819.00	0.99%	1.68%	1,835.91	1.96%	2.62%	1,836.53	2.66%
p14	2	1,789.01	1,804.37	0.21%	0.86%	1,819.00	0.99%	1.68%	1,835.91	1.96%	2.62%	1,836.53	2.66%
p15	4	3,969.33	3,974.73	0.14%	0.14%	3,996.33	0.68%	0.68%	4,023.33	1.34%	1.36%	4,204.07	5.91%
p16	4	3,969.33	3,974.73	0.14%	0.14%	3,996.33	0.68%	0.68%	4,023.33	1.34%	1.36%	4,204.07	5.91%
p17	4	3,969.33	3,974.73	0.14%	0.14%	3,996.33	0.68%	0.68%	4,023.33	1.34%	1.36%	4,204.07	5.91%
p18	6	6,859.44	6,867.47	0.12%	0.12%	6,902.84	0.52%	0.63%	6,931.78	1.24%	1.05%	7,395.74	7.82%
p19	6	6,859.44	6,867.47	0.12%	0.12%	6,902.84	0.52%	0.63%	6,931.78	1.24%	1.05%	7,395.74	7.82%
p20	6	7,124.91	7,290.24	0.21%	2.32%	7,323.05	0.82%	2.78%	7,349.98	1.50%	3.16%	7,875.24	10.53%
p21	9	11,603.57	11,674.90	0.21%	0.61%	11,754.81	0.65%	1.30%	11,933.12	1.54%	2.84%	12,472.04	7.48%
p22	9	11,603.57	11,674.90	0.21%	0.61%	11,754.81	0.65%	1.30%	11,933.12	1.54%	2.84%	12,472.04	7.48%
p23	9	11,603.57	11,674.90	0.21%	0.61%	11,754.81	0.65%	1.30%	11,933.12	1.54%	2.84%	12,472.04	7.48%
pr01	4	963.77	1,099.76	0.49%	14.11%	1,121.36	1.11%	16.35%	1,132.99	1.50%	17.56%	1,242.90	28.96%
pr02	4	1,505.47	1,560.76	2.12%	3.67%	1,695.59	4.69%	12.63%	1,743.32	3.50%	15.80%	1,767.52	17.41%
pr03	4	2,293.73	2,392.84	0.64%	4.32%	2,483.37	0.00%	8.27%	2,483.37	0.00%	8.27%	2,483.37	8.27%
pr04	4	2,614.08	2,724.61	1.26%	4.23%	2,818.66	2.93%	7.83%	2,829.81	3.25%	8.25%	3,008.51	15.09%
pr05	4	2,914.51	3,011.12	0.00%	3.31%	3,011.12	0.00%	3.31%	3,011.12	0.00%	3.31%	3,011.12	3.31%
pr06	4	3,793.28	3,909.61	0.80%	3.07%	3,974.39	0.00%	4.77%	3,974.39	0.00%	4.77%	3,974.39	4.77%
pr07	6	1,464.28	1,535.97	0.33%	4.90%	1,556.03	1.06%	6.27%	1,571.80	1.65%	7.34%	1,635.05	11.66%
pr08	6	2,149.61	2,371.91	0.97%	10.34%	2,431.35	3.91%	13.11%	2,474.35	3.35%	15.11%	2,715.37	26.32%
pr09	6	3,057.04	3,296.10	0.57%	7.82%	3,329.66	1.43%	8.92%	3,375.69	2.55%	10.42%	3,557.77	16.38%
pr10	6	3,898.41	4,174.38	0.75%	7.08%	4,218.67	0.71%	8.22%	4,223.25	0.54%	8.33%	4,254.24	9.13%
Avg.				0.63%	3.22%		1.10%	5.01%		1.18%	5.84%		8.71%

Table 2: Results obtained for the overlapping case and 2-day shift.

		Best-case Scenario with 100% flexibility	Intermediate Scenario with high flexibility (days = 1; penalty = Low)			Intermediate Scenario with medium flexibility (days = 1; penalty = Medium)			Intermediate Scenario with low flexibility (days = 1; penalty = High)			Worse-case Scenario with 0% flexibility	
Inst.	Ndays	OBS (1)	OBS COST (2)	Discount	GAP (2)-(1)	OBS COST (3)	Discount	GAP (3)-(1)	OBS COST (4)	Discount	GAP (4)-(1)	OBS COST (5)	GAP (5)-(1)
p01	4	746.64	763.99	2.04%	2.32%	817.82	7.76%	9.53%	837.29	1.49%	12.14%	840.02	12.51%
p02	4	529.72	557.95	2.46%	5.33%	606.75	7.60%	14.54%	639.78	3.28%	20.78%	641.37	21.08%
p03	5	850.46	883.46	2.67%	3.88%	900.04	2.99%	5.83%	921.53	1.09%	8.36%	922.39	8.46%
p04	2	1,171.61	1,175.71	0.35%	0.35%	1,191.87	0.00%	1.73%	1,191.87	0.00%	1.73%	1,191.87	1.73%
p05	2	830.05	834.25	0.50%	0.51%	836.94	0.18%	0.83%	838.44	0.36%	1.01%	838.96	1.07%
p06	3	1,171.36	1,172.56	0.10%	0.10%	1,177.36	0.51%	0.51%	1,183.36	1.01%	1.02%	1,190.42	1.63%
p07	4	1,184.42	1,202.61	1.72%	1.54%	1,242.09	0.40%	4.87%	1,247.09	0.80%	5.29%	1,264.13	6.73%
p08	2	5,307.75	5,332.73	0.31%	0.47%	5,358.73	0.17%	0.96%	5,364.12	0.34%	1.06%	5,381.24	1.38%
p09	3	4,515.76	4,668.76	1.45%	3.39%	4,803.94	1.85%	6.38%	4,849.41	2.06%	7.39%	4,864.68	7.73%
p10	4	5,281.13	5,428.90	0.74%	2.80%	5,516.89	0.22%	4.46%	5,530.99	0.43%	4.73%	5,532.13	4.75%
p11	5	5,367.76	5,458.04	1.28%	1.68%	5,531.91	0.28%	3.06%	5,547.41	0.56%	3.35%	5,625.84	4.81%
p12	2	1,789.01	1,804.37	0.21%	0.86%	1,819.00	0.99%	1.68%	1,835.91	1.96%	2.62%	1,836.53	2.66%
p13	2	1,789.01	1,804.37	0.21%	0.86%	1,819.00	0.99%	1.68%	1,835.91	1.96%	2.62%	1,836.53	2.66%
p14	2	1,789.01	1,804.37	0.21%	0.86%	1,819.00	0.99%	1.68%	1,835.91	1.96%	2.62%	1,836.53	2.66%
p15	4	3,969.33	3,974.73	0.14%	0.14%	3,996.33	0.68%	0.68%	4,023.33	1.34%	1.36%	4,204.07	5.91%
p16	4	3,969.33	3,974.73	0.14%	0.14%	3,996.33	0.68%	0.68%	4,023.33	1.34%	1.36%	4,204.07	5.91%
p17	4	3,969.33	3,974.73	0.14%	0.14%	3,996.33	0.68%	0.68%	4,023.33	1.34%	1.36%	4,204.07	5.91%
p18	6	6,859.44	6,867.47	0.12%	0.12%	6,902.84	0.52%	0.63%	6,931.78	1.24%	1.05%	7,395.74	7.82%
p19	6	6,859.44	6,867.47	0.12%	0.12%	6,902.84	0.52%	0.63%	6,931.78	1.24%	1.05%	7,395.74	7.82%
p20	6	7,124.91	7,290.24	0.21%	2.32%	7,323.05	0.82%	2.78%	7,349.98	1.50%	3.16%	7,875.24	10.53%
p21	9	11,603.57	11,629.43	0.10%	0.22%	11,754.81	0.65%	1.30%	11,933.12	1.54%	2.84%	12,472.04	7.48%
p22	9	11,603.57	11,629.43	0.10%	0.22%	11,754.81	0.65%	1.30%	11,933.12	1.54%	2.84%	12,472.04	7.48%
p23	9	11,603.57	11,629.43	0.10%	0.22%	11,754.81	0.65%	1.30%	11,933.12	1.54%	2.84%	12,472.04	7.48%
pr01	4	963.77	1,020.10	1.08%	5.84%	1,071.17	3.29%	11.14%	1,092.49	3.34%	13.36%	1,242.90	28.96%
pr02	4	1,505.47	1,543.26	2.17%	2.51%	1,639.10	3.42%	8.88%	1,709.33	6.67%	13.54%	1,767.52	17.41%
pr03	4	2,293.73	2,332.84	1.63%	1.71%	2,425.23	2.59%	5.73%	2,480.73	1.94%	8.15%	2,483.37	8.27%
pr04	4	2,614.08	2,706.66	1.04%	3.54%	2,792.76	3.16%	6.84%	2,829.81	3.25%	8.25%	3,008.51	15.09%
pr05	4	2,914.51	2,966.27	0.87%	1.78%	2,997.05	0.78%	2.83%	3,011.12	0.00%	3.31%	3,011.12	3.31%
pr06	4	3,793.28	3,909.61	0.80%	3.07%	3,974.39	0.00%	4.77%	3,974.39	0.00%	4.77%	3,974.39	4.77%
pr07	6	1,464.28	1,474.41	0.69%	0.69%	1,514.91	3.34%	3.46%	1,551.20	2.29%	5.94%	1,635.05	11.66%
pr08	6	2,149.61	2,342.97	2.69%	9.00%	2,416.98	5.17%	12.44%	2,472.91	4.44%	15.04%	2,715.37	26.32%
pr09	6	3,057.04	3,161.99	1.12%	3.43%	3,252.09	2.68%	6.38%	3,312.98	3.86%	8.37%	3,557.77	16.38%
pr10	6	3,898.41	4,162.40	0.81%	6.77%	4,214.32	0.53%	8.10%	4,223.25	0.54%	8.33%	4,254.24	9.13%
Avg.				0.86%	2.03%		1.69%	4.19%		1.71%	5.51%		8.71%

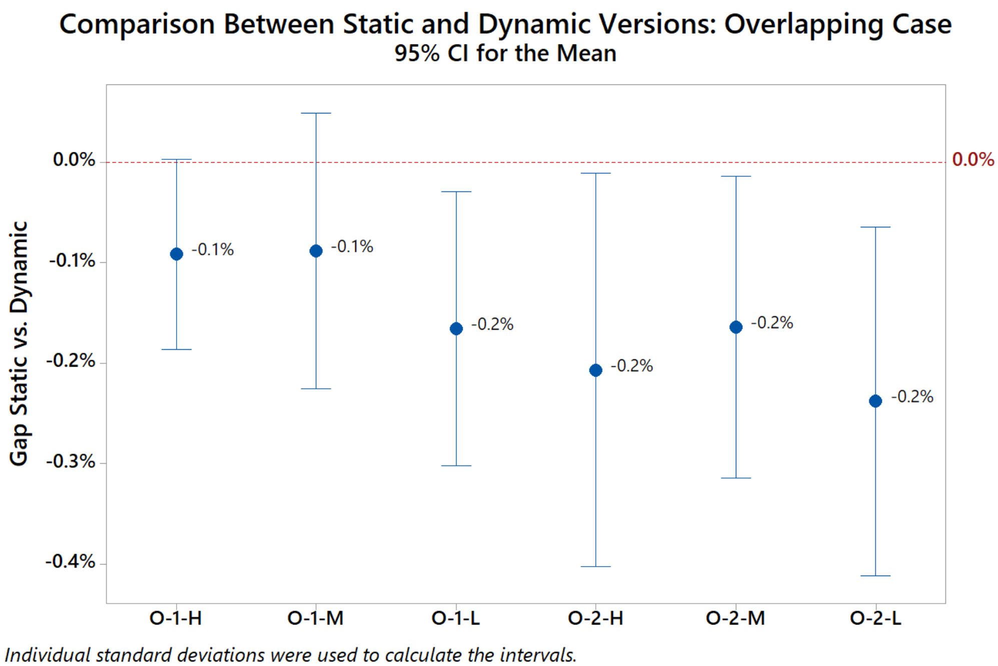

Figure 5: Comparison between the depot-based (static) and the dynamic-nearest schemes.

employed, although it seems that the *dynamic-nearest* scheme is the one providing the best average results. This similarity of results among the different schemes is also explained by the fact that we also use a local search in the algorithm, which promotes that geographically-clustered customers will be serviced by the same vehicle (as far as the penalty cost associated with potential re-allocations is lower than its associated benefits).

Similarly to what we did in the case of the geographically dispersed depots, it is possible to compare, for the co-located case with overlapping depots, how the distribution costs vary as the level of flexibility is modified. Figure 6 illustrates how a reduction in the level of flexibility leads to a higher gap with respect to the full-flexibility scenario and, accordingly, to an increase in total distribution costs. Thus, while the aforementioned gap is about 2% for the high-flexibility with 2-day shift scenario (O-2-H), it raises up to about 9% for the zero-flexibility scenario (O-Z). In summary, our approach shows that, at least for the two sets of data considered (non-overlapping and overlapping cases), companies should consider adding flexibility to their MPVRP in order to obtain notable reductions in their distribution cost. This flexibility can be gained by negotiating reasonably-affordable discounts in exchange for the possibility of shifting deliveries from one day to another.

Table 3: Comparison among the three desirability schemes for a 1-day shift.

		Intermediate Scenario with high flexibility (days = 1; penalty = low) (dB, dN, kN)				Intermediate Scenario with medium flexibility (days = 1; penalty = medium) (dB, dN, kN)				Intermediate Scenario with low flexibility (days = 1; penalty = high) (dB, dN, kN)			
Instance	Ndays	OBS	GAP dB-OBS	GAP dN-OBS	GAP kN-OBS	OBS	GAP dB-OBS	GAP dN-OBS	GAP kN-OBS	OBS	GAP dB-OBS	GAP dN-OBS	GAP kN-OBS
p01	4	803.64	0.23%	0.00%	0.55%	834.43	0.00%	0.00%	0.00%	838.43	0.19%	0.19%	0.00%
p02	4	577.97	0.00%	0.00%	0.24%	624.89	0.00%	0.00%	0.00%	639.78	0.25%	0.00%	0.00%
p03	5	896.51	0.00%	0.00%	0.74%	916.53	0.00%	0.00%	0.00%	919.10	0.26%	0.26%	0.00%
p04	2	1,175.71	0.00%	0.00%	1.00%	1,191.87	0.00%	0.00%	0.00%	1,191.87	0.00%	0.00%	0.00%
p05	2	834.00	0.03%	0.10%	0.00%	836.68	0.03%	0.22%	0.00%	838.18	0.03%	0.09%	0.00%
p06	3	1,172.56	0.00%	0.01%	0.35%	1,177.36	0.00%	0.01%	0.35%	1,183.36	0.00%	0.01%	0.34%
p07	4	1,213.55	0.00%	0.00%	0.47%	1,241.22	0.07%	0.07%	0.00%	1,247.09	0.00%	0.00%	0.00%
p08	2	5,331.16	0.03%	0.00%	0.61%	5,347.91	0.20%	0.00%	0.40%	5,364.12	0.00%	0.07%	0.23%
p09	3	4,721.35	0.00%	0.00%	0.00%	4,823.23	0.00%	0.03%	0.21%	4,864.68	0.00%	0.00%	0.00%
p10	4	5,466.09	0.21%	0.00%	0.21%	5,516.89	0.00%	0.19%	0.28%	5,531.59	0.01%	0.00%	0.01%
p11	5	5,447.78	0.00%	0.09%	0.09%	5,531.91	0.00%	0.04%	0.04%	5,547.41	0.00%	0.10%	0.10%
p12	2	1,791.23	0.73%	0.73%	0.00%	1,795.23	1.32%	1.33%	0.00%	1,800.23	1.98%	1.92%	0.00%
p13	2	1,791.23	0.73%	0.73%	0.00%	1,795.23	1.32%	1.33%	0.00%	1,800.23	1.98%	1.92%	0.00%
p14	2	1,791.23	0.73%	0.73%	0.00%	1,795.23	1.32%	1.33%	0.00%	1,800.23	1.98%	1.92%	0.00%
p15	4	3,974.73	0.00%	0.00%	0.11%	3,996.33	0.00%	0.21%	0.29%	4,019.98	0.08%	0.00%	0.60%
p16	4	3,974.73	0.00%	0.00%	0.11%	3,996.33	0.00%	0.21%	0.29%	4,019.98	0.08%	0.00%	0.60%
p17	4	3,974.73	0.00%	0.00%	0.11%	3,996.33	0.00%	0.21%	0.29%	4,019.98	0.08%	0.00%	0.60%
p18	6	6,867.39	0.00%	0.00%	0.00%	6,901.33	0.02%	0.00%	0.02%	6,931.78	0.00%	0.15%	0.15%
p19	6	6,867.39	0.00%	0.00%	0.00%	6,901.33	0.02%	0.00%	0.02%	6,931.78	0.00%	0.15%	0.15%
p20	6	7,284.73	0.08%	0.00%	0.08%	7,297.74	0.35%	0.00%	0.35%	7,344.24	0.08%	0.00%	0.08%
p21	9	11,674.90	0.00%	0.00%	0.00%	11,754.81	0.00%	0.00%	0.00%	11,796.47	1.16%	0.00%	1.16%
p22	9	11,674.90	0.00%	0.00%	0.00%	11,754.81	0.00%	0.00%	0.00%	11,796.47	1.16%	0.00%	1.16%
p23	9	11,674.90	0.00%	0.00%	0.00%	11,754.81	0.00%	0.00%	0.00%	11,796.47	1.16%	0.00%	1.16%
pr01	4	1,099.76	0.00%	0.00%	0.00%	1,120.57	0.07%	0.00%	0.07%	1,132.99	0.00%	0.00%	0.00%
pr02	4	1,560.76	0.00%	0.00%	0.00%	1,678.28	1.03%	0.61%	0.00%	1,723.12	1.17%	0.00%	0.52%
pr03	4	2,392.84	0.00%	0.00%	0.40%	2,455.78	1.12%	0.00%	1.12%	2,483.37	0.00%	0.00%	0.00%
pr04	4	2,724.61	0.00%	0.10%	0.10%	2,768.22	1.82%	0.00%	1.82%	2,819.22	0.38%	0.00%	0.38%
pr05	4	3,011.12	0.00%	0.00%	0.00%	3,011.12	0.00%	0.00%	0.00%	3,011.12	0.00%	0.00%	0.00%
pr06	4	3,904.35	0.13%	0.00%	0.13%	3,974.39	0.00%	0.00%	0.00%	3,974.39	0.00%	0.00%	0.00%
pr07	6	1,535.97	0.00%	0.03%	0.00%	1,555.97	0.00%	0.00%	0.00%	1,567.42	0.28%	0.00%	0.00%
pr08	6	2,367.71	0.18%	0.18%	0.00%	2,431.35	0.00%	0.03%	0.08%	2,474.35	0.00%	0.08%	0.01%
pr09	6	3,257.37	1.19%	0.00%	1.19%	3,329.66	0.00%	0.14%	0.77%	3,374.10	0.05%	0.00%	0.59%
pr10	6	4,136.23	0.92%	0.00%	0.92%	4,209.31	0.22%	0.00%	0.22%	4,222.81	0.01%	0.00%	0.01%
Avg.			0.16%	0.08%	0.22%		0.27%	0.18%	0.20%		0.38%	0.21%	0.24%

Table 4: Comparison among the three desirability scheme for a 2-day shift.

		Intermediate Scenario with high flexibility (days = 2; penalty = low) (dB, dN, kN)				Intermediate Scenario with medium flexibility (days = 2; penalty = medium) (dB, dN, kN)				Intermediate Scenario with low flexibility (days = 2; penalty = high) (dB, dN, kN)			
Instance	Ndays	OBS	GAP dB-OBS	GAP dN-OBS	GAP kN-OBS	OBS	GAP dB-OBS	GAP dN-OBS	GAP kN-OBS	OBS	GAP dB-OBS	GAP dN-OBS	GAP kN-OBS
p01	4	763.99	0.00%	0.01%	2.26%	817.82	0.00%	0.00%	1.81%	837.29	0.00%	0.00%	0.00%
p02	4	552.63	0.96%	0.00%	0.00%	606.75	0.00%	0.00%	0.00%	639.78	0.00%	0.00%	0.00%
p03	5	868.90	1.68%	0.76%	0.00%	900.04	0.00%	0.00%	0.21%	919.10	0.26%	0.26%	0.00%
p04	2	1,175.71	0.00%	0.00%	1.00%	1,191.87	0.00%	0.00%	0.00%	1,191.87	0.00%	0.00%	0.00%
p05	2	834.00	0.03%	0.10%	0.00%	836.68	0.03%	0.22%	0.00%	838.18	0.03%	0.09%	0.00%
p06	3	1,172.56	0.00%	0.01%	0.35%	1,177.36	0.00%	0.01%	0.35%	1,183.36	0.00%	0.01%	0.34%
p07	4	1,201.27	0.11%	0.00%	1.49%	1,241.22	0.07%	0.07%	0.00%	1,247.09	0.00%	0.00%	0.00%
p08	2	5,331.16	0.03%	0.00%	0.61%	5,347.91	0.20%	0.00%	0.40%	5,364.12	0.00%	0.07%	0.23%
p09	3	4,665.50	0.07%	0.24%	0.00%	4,757.49	0.98%	0.00%	1.01%	4,807.49	0.87%	0.00%	0.94%
p10	4	5,416.35	0.23%	0.00%	1.13%	5,516.89	0.00%	0.19%	0.28%	5,530.99	0.00%	0.01%	0.02%
p11	5	5,452.72	0.10%	0.00%	0.00%	5,531.91	0.00%	0.04%	0.04%	5,547.41	0.00%	0.10%	0.10%
p12	2	1,791.23	0.73%	0.73%	0.00%	1,795.23	1.32%	1.33%	0.00%	1,800.23	1.98%	1.92%	0.00%
p13	2	1,791.23	0.73%	0.73%	0.00%	1,795.23	1.32%	1.33%	0.00%	1,800.23	1.98%	1.92%	0.00%
p14	2	1,791.23	0.73%	0.73%	0.00%	1,795.23	1.32%	1.33%	0.00%	1,800.23	1.98%	1.92%	0.00%
p15	4	3,974.73	0.00%	0.00%	0.11%	3,996.33	0.00%	0.06%	0.29%	4,019.98	0.08%	0.00%	0.27%
p16	4	3,974.73	0.00%	0.00%	0.11%	3,996.33	0.00%	0.00%	0.29%	4,019.98	0.08%	0.00%	0.27%
p17	4	3,974.73	0.00%	0.00%	0.11%	3,996.33	0.00%	0.00%	0.29%	4,019.98	0.08%	0.00%	0.27%
p18	6	6,867.39	0.00%	0.00%	0.00%	6,899.90	0.04%	0.00%	0.04%	6,931.78	0.00%	0.15%	0.15%
p19	6	6,867.39	0.00%	0.00%	0.00%	6,899.90	0.04%	0.00%	0.04%	6,931.78	0.00%	0.15%	0.15%
p20	6	7,284.73	0.08%	0.00%	0.08%	7,297.74	0.35%	0.00%	0.35%	7,344.24	0.08%	0.00%	0.08%
p21	9	11,629.43	0.00%	0.00%	0.39%	11,754.81	0.00%	0.00%	0.00%	11,796.47	1.16%	0.00%	1.16%
p22	9	11,629.43	0.00%	0.00%	0.39%	11,754.81	0.00%	0.00%	0.00%	11,796.47	1.16%	0.00%	1.16%
p23	9	11,629.43	0.00%	0.00%	0.39%	11,754.81	0.00%	0.00%	0.00%	11,796.47	1.16%	0.00%	1.16%
pr01	4	1,020.10	0.00%	0.00%	0.59%	1,071.17	0.00%	0.00%	4.69%	1,092.49	0.00%	0.00%	3.71%
pr02	4	1,541.59	0.11%	0.00%	1.24%	1,629.79	0.57%	0.00%	2.98%	1,686.44	1.36%	0.00%	2.71%
pr03	4	2,332.84	0.00%	0.11%	0.61%	2,425.23	0.00%	0.00%	2.40%	2,480.73	0.00%	0.11%	0.11%
pr04	4	2,692.57	0.52%	0.00%	1.30%	2,768.22	0.89%	0.00%	1.75%	2,819.22	0.38%	0.00%	0.38%
pr05	4	2,966.27	0.00%	0.07%	1.51%	2,997.05	0.00%	0.05%	0.47%	3,011.12	0.00%	0.00%	0.00%
pr06	4	3,833.93	1.97%	0.00%	1.97%	3,894.71	2.05%	0.00%	2.05%	3,957.14	0.44%	0.00%	0.44%
pr07	6	1,474.41	0.00%	0.00%	0.86%	1,514.91	0.00%	0.00%	0.00%	1,551.20	0.00%	0.00%	0.00%
pr08	6	2,342.97	0.00%	0.00%	1.06%	2,405.76	0.47%	0.00%	0.81%	2,432.82	1.65%	0.00%	1.72%
pr09	6	3,161.99	0.00%	0.00%	4.24%	3,240.97	0.34%	0.00%	3.52%	3,312.98	0.00%	0.10%	2.44%
pr10	6	4,066.79	2.35%	0.00%	2.65%	4,209.31	0.12%	0.00%	0.22%	4,222.81	0.01%	0.00%	0.01%
Avg.			0.32%	0.11%	0.74%		0.31%	0.14%	0.74%		0.45%	0.21%	0.54%

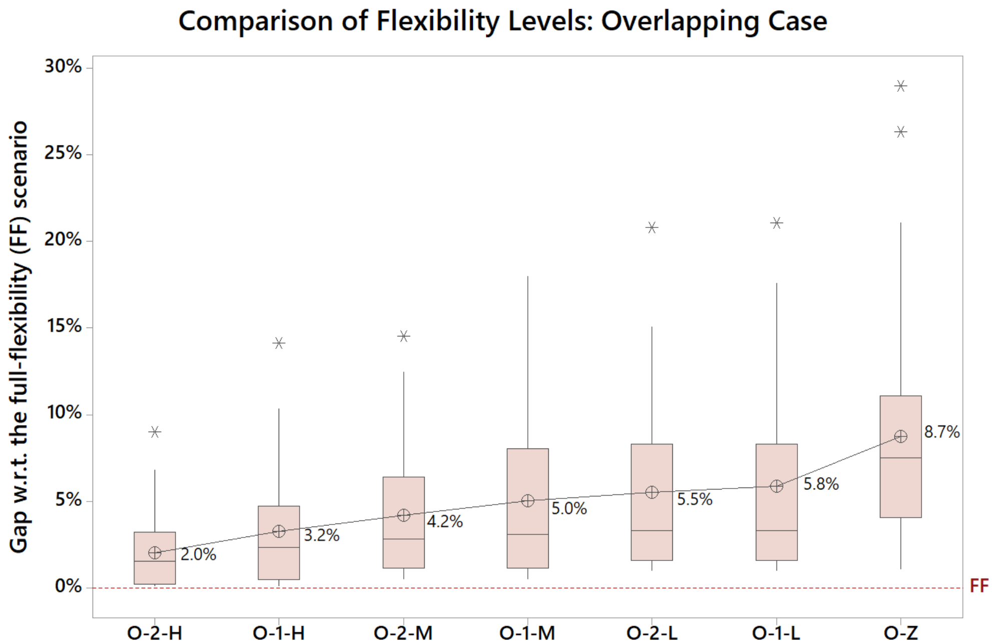

Figure 6: A visual comparison of the different flexibility scenarios for the co-located case.

5. Conclusions

This paper has introduced a new realistic variant of the multi-period vehicle routing problem in which price discounts are given to customers in exchange for some flexibility in the delivery times. In particular, the carrier reduces its distribution costs by advancing or delaying one or two days the service of some selected customers and, at the same time, the reallocated customers benefit from a discount fee for their flexibility. In order to solve this problem, a metaheuristic approach has been proposed. Our algorithm combines an iterated local search framework with biased-randomized techniques that are applied both during the allocation of customers to days as well as during the subsequent routing process. It is a two-stage approach. Firstly, a round-robin process is combined with biased-randomization for fast generation of 'balanced' customers-to-days allocation maps. In this step, three different desirability schemes are proposed to establish a priority list in the allocation of customers per depot. Each of the allocation maps generated is then routed using a fast heuristic, which provides an initial estimate of its quality as potential delivery plan. The most promising maps are then selected and processed by a more intensive and time-consuming routing algorithm, which provides final assignment-and-routing solutions of higher quality.

The algorithm described has been tested using a series of well-known benchmarks for the multi-depot vehicle routing problem. As discussed in the paper, the multi-period variant analyzed could be considered as a particular case of multi-depot vehicle routing problem. By employing these well-known benchmarks, it is possible to validate –at least to some extent– the performance efficiency of our approach. In a second experiment, a series of derived instances are also employed to extend our analysis to a more realistic case with co-located depots (i.e., the geographic coordinates of the depot are fixed regardless of the day considered). All in all, our approach is able to quantify the potential benefits that the manager could reach by applying different levels of flexibility in the delivery day and discount fee. From a managerial perspective, this information can be extremely useful to negotiate the more convenient flexibility conditions (timing and discount fees) with their customers.

As future work, we plan to extend our work in different ways: *(i)* by incorporating a probabilistic customer's behavior –i.e., customers will accept the offered discount with a given probability; and *(ii)* by including reactive policies that can allow the algorithm to efficiently response to dynamic events such as order cancellations or arrival of new orders.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Education, Culture, and Sports via a Jose Castillejo grant (CAS16/00201).

References

References

Albareda-Sambola, M., Fernández, E., & Laporte, G. (2014). The dynamic multiperiod vehicle routing problem with probabilistic information. *Computers and Operations Research*, *48*, 31–39.

Alonso, F., Alvarez, M. J., & Beasley, J. E. (2008). A tabu search algorithm for the periodic vehicle routing problem with multiple vehicle trips and accessibility restrictions. *Journal of the Operational Research Society*, *59*, 963–976.

Angelelli, E., & Speranza, M. G. (2002). The periodic vehicle routing problem with intermediate facilities. *European journal of Operational research*, *137*, 233–247.

Archetti, C., Fernández, E., & Huerta-Muñoz, D. L. (2017). The flexible periodic vehicle routing problem. *Computers & Operations Research*, *85*, 58–70.

Archetti, C., Jabali, O., & Speranza, M. G. (2015). Multi-period Vehicle Routing Problem with Due dates. *Computers and Operations Research*, *61*, 122–134.

Beltrami, E. J., Bodin, L. D., & Brook, S. (1974). Networks and Vehicle Routing for Municipal Waste Collection. *Networks*, *4*, 65–94.

Cacchiani, V., Hemmelmayr, V. C., & Tricoire, F. (2014). A set-covering based heuristic algorithm for the periodic vehicle routing problem. *Discrete Applied Mathematics*, *163*, 53–64.

Caceres-Cruz, J., Arias, P., Guimarans, D., Riera, D., & Juan, A. A. (2015). Rich vehicle routing problem: Survey. *ACM Computing Surveys (CSUR)*, *47*, 32.

Christofides, N., & Beasley, J. E. (1984). The period routing problem. *Networks*, *14*, 237–256.

Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, *12*, 568–581.

Cordeau, J.-F., Gendreau, M., & Laporte, G. (1997). A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, *30*, 105–119.

Dominguez, O., Guimarans, D., Juan, A. A., & de la Nuez, I. (2016a). A biased-randomised large neighbourhood search for the two-dimensional vehicle routing problem with backhauls. *European Journal of Operational Research*, *255*, 442–462.

Dominguez, O., Juan, A. A., Barrios, B., Faulin, J., & Agustin, A. (2016b). Using biased randomization for solving the two-dimensional loading vehicle routing problem with heterogeneous fleet. *Annals of Operations Research*, *236*, 383–404.

Francis, P., Smilowitz, K., & Tzur, M. (2006). The period vehicle routing problem with service choice. *Transportation science*, *40*, 439–454.

Francis, P., Smilowitz, K., & Tzur, M. (2007). Flexibility and complexity in periodic distribution problems. *Naval Research Logistics (NRL)*, *54*, 136–150.

Francis, P. M., Smilowitz, K. R., & Tzur, M. (2008). The period vehicle routing problem and its extensions. In *The vehicle routing problem: latest advances and new challenges* (pp. 73–102). Springer.

Grasas, A., Juan, A. A., Faulin, J., De Armas, J., & Ramalhinho, H. (2017). Biased randomization of heuristics using skewed probability distributions: a survey and some applications. *Computers & Industrial Engineering*, *110*, 216–228.

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*, 791–802.

Juan, A. A., Faulin, J., Ferrer, A., Lourenço, H. R., & Barrios, B. (2013). Mirha: multi-start biased randomization of heuristics with adaptive local search for solving non-smooth routing problems. *Top*, *21*, 109–132.

Juan, A. A., Faulin, J., Jorba, J., Riera, D., Masip, D., & Barrios, B. (2011). On the use of monte carlo simulation, cache and splitting techniques to improve the clarke and wright savings heuristics. *Journal of the Operational Research Society*, *62*, 1085–1097.

Juan, A. A., Lourenço, H. R., Mateo, M., Luo, R., & Castella, Q. (2014). Using iterated local search for solving the flow-shop problem: Parallelization, parametrization, and randomization issues. *International Transactions in Operational Research*, *21*, 103–126.

Li, J., Pan, Q., & Xie, S. (2012). An effective shuffled frog-leaping algorithm for multi-objective flexible job shop scheduling problems. *Applied Mathematics and Computation*, *218*, 9353–9371.

Lourenço, H. R., Martin, O. C., & Stützle, T. (2010). Iterated local search: Framework and applications. In *Handbook of metaheuristics* (pp. 363–397). Springer.

Luo, Z., Qin, H., Che, C., & Lim, A. (2015). On service consistency in multi-period vehicle routing. *European Journal of Operational Research*, *243*, 731–744.

Mourgaya, M., & Vanderbeck, F. (2007). Column generation based heuristic for tactical planning in multi-period vehicle routing. *European Journal of Operational Research*, *183*, 1028–1041.

Pacheco, J., Alvarez, A., García, I., & Angel-Bello, F. (2012). Optimizing vehicle routes in a bakery company allowing flexibility in delivery dates. *Journal of the Operational Research Society*, *63*, 569–581.

Pisinger, D., & Ropke, S. (2007). A general heuristic for vehicle routing problems. *Computers & Operations Research*, *34*, 2403 – 2435.

Rahimi-Vahed, A., Crainic, T. G., Gendreau, M., & Rei, W. (2013). A path relinking algorithm for a multi-depot periodic vehicle routing problem. *Journal of Heuristics*, *19*, 497–524.

Russell, R., & Igo, W. (1979). An assignment routing problem. *Networks*, *9*, 1–17.

Russell, R. A. (1977). Technical note—an effective heuristic for the m-tour traveling salesman problem with some side conditions. *Operations Research*, *25*, 517–524.

Savelsbergh, M., & Van Woensel, T. (2016). 50th anniversary invited article—city logistics: Challenges and opportunities. *Transportation Science*, *50*, 579–590.

Toth, P., & Vigo, D. (2014). *Vehicle routing: problems, methods, and applications*. SIAM.

Vidal, T., Crainic, T. G., Gendreau, M., Lahrichi, N., & Rei, W. (2012). A Hybrid Genetic Algorithm for Multidepot and Periodic Vehicle Routing Problems. *Operations Research*, *60*, 611–624.

Wen, M., Cordeau, J. F., Laporte, G., & Larsen, J. (2010). The dynamic multi-period vehicle routing problem. *Computers and Operations Research*, *37*, 1615–1623.

Zäpfel, G., & Bögl, M. (2008). Multi-period vehicle routing and crew scheduling with outsourcing options. *International Journal of Production Economics*, *113*, 980–996.

Zhang, Z., Che, O., Cheang, B., Lim, A., & Qin, H. (2013). A memetic algorithm for the multiperiod vehicle routing problem with profit. *European Journal of Operational Research*, *229*, 573–584.