Route and fleet design for cyclic inventory routing

Raa, Birger; Dullaert, Wout

published in
European Journal of Operational Research
2017

DOI (link to publisher)
10.1016/j.ejor.2016.06.009

document version
Publisher's PDF, also known as Version of record

European Journal of Operational Research 256 (2017) 404–411

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Production, Manufacturing and Logistics

Route and fleet design for cyclic inventory routing

Birger Raa[a,b,*], Wout Dullaert[c]

[a] Department of Industrial Systems Engineering and Product Design, Ghent University, Technologiepark 903, Zwijnaarde 9052, Belgium
[b] Antwerp Maritime Academy, Belgium
[c] Department of Information, Logistics and Innovation, Vrije Universiteit Amsterdam, Netherlands

ARTICLE INFO

Article history:
Received 12 August 2015
Accepted 6 June 2016
Available online 11 June 2016

Keywords:
Inventory routing
Cyclic planning
Delivery scheduling
Periodic routing
Distribution logistics

ABSTRACT

This paper presents a novel solution approach for planning cyclic distribution from a single depot to multiple customers with constant, deterministic demand rates. The objective is to minimize the total cost rate consisting of fleet costs, distribution costs from the depot to the customers and inventory holding costs at the customers. A solution is built in two phases: designing routes and composing the fleet. When designing vehicle routes in the first phase, the route cycle times are chosen such that the distribution and inventory holding costs are minimized. When assigning routes to vehicles in the second phase, the routes remain unchanged, but their cycle times can be adjusted to minimize the required number of vehicles. The building blocks of this two-phase solution approach are embedded in a metaheuristic framework. Computational experiments show that the resulting solution framework outperforms existing solution approaches.

1. Introduction

'Vendor Managed Inventory' (VMI) or 'Supplier Managed Inventory' (SMI) aims at seamlessly integrating distribution and inventory management in a two-echelon supply chain. To do so, customers share the information about their demand and inventory levels with the supplier. This allows the latter to aggregate and integrate decisions about which customers to replenish at what time with decisions about the design of the vehicle routes for making these replenishments. This results in an integrated optimization problem known as the 'Inventory Routing Problem' (IRP), which recently has received an increasing amount of research interest. An overview of the academic literature on the IRP and a typology of IRPs is presented in Andersson, Hoff, Christiansen, Hasle, and Lokketangen (2010), and more recently in Coelho, Cordeau, and Laporte (2014).

This paper focuses on an inventory routing problem with an infinite time horizon, deterministic demand, a one-to-many topology, and a homogeneous fleet according to the classification of Andersson et al. (2010). We consider a set of customers that has to be replenished from a single depot. Every customer has a given, constant demand rate and must be cyclically replenished, i.e., the time between consecutive replenishments, called 'cycle time', must always be the same. In the depot, vehicles are available to perform the replenishments. Vehicles can combine replenishments by visiting multiple customers together in a route. Therefore, a so-called 'fixed partition policy' (FPP) is adopted. In a FPP, the set of customers is partitioned into subsets and for every subset, a separate, minimum-cost route is designed. The cycle times of the routes are then chosen such that the resulting cost rate, i.e., the average cost per day, consisting of inventory holding costs at the customers plus the route costs, is minimized.

Most of the literature on cyclic IRPs also adopts FPPs, often based on geographical proximity. FPPs were introduced by Anily and Federgruen (1990) and thoroughly analyzed by Anily and Bramel (2004), and Chan, Speranza, and Bertazzi (2013) amongst others. However, these authors do not explicitly consider the fixed costs of the vehicles, nor do they determine the required fleet size. They merely assume that sufficient vehicles are available, i.e., one for each partition. From a long-term perspective, however, the fleet size and its resulting cost rate cannot be assumed to be given, but should also be considered as a decision variable. Thus, solving the cyclic IRP comprises two major interrelated subproblems, namely designing a set of efficient routes and then deciding on the required vehicle fleet that is needed to cyclically repeat these routes. This paper presents a solution approach for the cyclic IRP that is based on this decomposed view. The solution approach first designs a set of routes covering all customers, similar to earlier work on FPPs, but then in a second phase, a schedule is constructed that

* Corresponding author at: Department of Industrial Systems Engineering and Product Design, Ghent University, Technologiepark 903, 9052 Zwijnaarde, Belgium. Tel.: +32 92645508.
E-mail addresses: birger.raa@ugent.be (B. Raa), wout.dullaert@vu.nl (W. Dullaert).

http://dx.doi.org/10.1016/j.ejor.2016.06.009
0377-2217/

assigns the routes to vehicles in such a way that the number of vehicles is minimized, an aspect of cyclic IRP that is overlooked in most previous research. Further, this two-phase approach is then used in a metaheuristic framework.

The remaining of this work is organized as follows. In Section 2, a brief review of relevant publications from the literature is provided and the contribution of this paper is stated. The two-phase solution methodology and the overall metaheuristic framework are discussed in Sections 3. The computational results in Sections 4, based on a set of benchmark instances, illustrate its performance. Finally, Section 5 provides conclusions and recommendations for additional research.

2. Literature review

As mentioned in the introduction above, the 'Inventory Routing Problem' has received considerable interest in the recent academic literature. Various versions of the IRP are being considered, differing in terms of planning horizon, demand characteristics, network topology, inventory policies, and solution methodologies. For the infinite horizon situation that is considered in this paper, the survey of Andersson et al. (2010) points out that fleet sizing and coordination of route frequencies is usually overlooked (Andersson et al., 2010, p. 1529). In this section, we give a chronological overview of the limited set of papers in the IRP literature that do address fleet sizing issues. We then identify the limitations of these papers and show how we contribute to this literature.

Larson (1988) introduced the so-called Strategic Inventory Routing Problem (SIRP), in which the objective is to minimize the required vessel fleet size for bringing waste from water treatment out into the ocean. Constraints related to limited storage capacity, limited vehicle capacity and transportation characteristics are taken into account. A heuristic savings algorithm, SIRSA, is presented, which assigns customers to clusters and visits all customers of a cluster in a single route. The objective is to minimize the cumulative vessel utilization across all routes, which reflects the actual number of vessels required. Webb and Larson (1995) generalize Larson's work by introducing the concept of *routesets*. A routeset consists of a number of component routes arranged in a specific order. The objective is then to minimize the cumulative vessel utilization across all routesets. Computational results show that this approach generally yields significant reductions in average vehicle requirement. However, still no actual fleet schedule is developed that assigns each iteration of any individual route to a particular vessel. It is only assumed that the strategic decision on the number of vessels will suffice to obtain feasible solutions for the eventual tactical/operational scheduling problems.

Campbell and Hardin (2005) consider the problem of minimizing the number of vehicles required to make strictly periodic delivery routes, i.e., where the time between iterations of a route is always exactly the same. At first, the assumption is made that a vehicle can only cover one trip per day. The complexity of this problem is analyzed and it is shown that the problem and many of its variants are NP-hard. A greedy algorithm is presented that can be used for any instance of the problem and gives optimal solutions for some special cases. Next, the problem is generalized by including the possibility of doing multiple trips per day for a vehicle. This extension drastically increases the complexity due to the resulting bin packing structure. The proposed greedy algorithm is modified by making use of a first-fit rule, similar to the well-known greedy bin packing heuristic.

To the best of our knowledge, the first paper on the cyclic inventory routing problem that considers fleet sizing by allocating routes to vehicles and taking into account a fixed cost rate for every vehicle required, is (Aghezzaf, Raa, & Landeghem, 2006). In that paper, customers are clustered into subsets, and a *multi-tour* is determined for each cluster, i.e., a collection of routes that are all performed by the same vehicle with the same cycle time. A heuristic column generation approach is presented, in which the columns are multi-tours generated by a modified Clarke-and-Wright heuristic, and the master problem is to cover all customers at minimal cost with a selection of multi-tours. In a later paper, Raa and Aghezzaf (2008) extend the multi-tours to *distribution patterns*, i.e., a collection of routes that are all performed by the same vehicle, but with possibly different cycle times. All routes of a distribution pattern have a cycle time that is an integer multiple of a base cycle time. That paper and the previous assume that cycle times are continuous values (e.g., 37.16 hours) which limits practical applicability. To address this, a more practical approach is developed in Raa and Aghezzaf (2009). That paper considers distribution patterns in which route cycle times have to be an integer number of days, and the daily vehicle driving time is restricted.

This paper contributes to the existing literature both in the generalization of the problem and in elaborating heuristic procedures that are more sophisticated (and less greedy) than the construction heuristics presented thus far. When compared to Webb and Larson (1995), the required fleet size is determined based on an actual cyclic fleet schedule, instead of a cumulative utilization measure. In comparison to Campbell and Hardin (2005), the problem is generalized by allowing the frequency of the routes to be adjusted in order to reduce the required fleet size. In Raa and Aghezzaf (2008); 2009), routes are constructed as part of a distribution pattern, and hence are already tied to a vehicle. A separate schedule per vehicle is then made with a greedy insertion heuristic. In our approach, routes are constructed first and allocated to vehicles afterwards. This allows for more flexibility in the route-to-vehicle allocations since different iterations of a route could be done by different vehicles. This generalization of the problem creates the potential for finding more cost efficient solutions, but increases the complexity of finding these solutions.

3. Solution approach

Given the complexity of the cyclic IRP, our aim was to develop a metaheuristic solution approach that outperforms existing approaches. Key considerations for designing the approach were: (i) using a flexible metaheuristic framework with proven performance; (ii) developing a simple strategy for guiding the search by balancing model components; and (iii) computational efficiency by exploiting information obtained during the search. These key considerations are in line with findings in the recent work of Vidal, Crainic, Gendreau, and Prins (2013) on 64 successful metaheuristics for 15 well-studied multi-attribute Vehicle Routing Problems (VRPs). Their analyses showed that Tabu search, adaptive large neighborhood search and especially hybrid genetic algorithms provide the framework for high performance solution approaches. They also conclude that the power of any state-of-the-art approach cannot be attributed to a single factor but to the extent in which the different components of a methodology are combined to intensify and diversify the search. Finally, they draw attention to the fact that many algorithms dedicate most of their computation time on evaluating moves and decisions during the search process without taking advantage of the calculations already performed during the search.

The solution approach presented in this Section was motivated by these considerations. The following sections present the details of the different building blocks of our solution approach. First, the route design subproblem is tackled in Section 3.1. Then, the subsequent fleet design problem is considered in Section 3.2. Finally, Section 3.3 explains the metaheuristic framework that is built around these elements.

3.1. Route design

In the route design phase, cyclic routes have to be constructed. For every route, not only the customer sequencing to minimize travel distance and cost has to be determined, but also the cycle time, i.e., the time between consecutive iterations of the route. This cycle time is determined by making the trade-off between the costs of making the route and the costs of holding inventory at the customers being replenished along the route. Also, the route cycle time is constrained by the vehicle loading capacity and the customer storage capacities.

Suppose set S is the set of all customers to be replenished and route r visits a subset S_r of these customers. Every customer $i \in S_r$ has a constant demand rate of d_i units per day and a storage capacity of κ_i units. A truck with a capacity of κ units is available for making the route. The cycle time of this route, denoted T_r (in days), has to be determined.

The customers in the route are all visited once every T_r days, meaning that customer $i \in S_r$ receives a delivery of $d_i \cdot T_r$ units. This delivery quantity is limited by the storage capacity κ_i of the customer, and therefore, the cycle time T_r must be less than or equal to $\frac{\kappa_i}{d_i}$. Further, the cycle time T_r is also restricted by the vehicle capacity. We assume a homogeneous fleet of vehicles with a loading capacity of κ units. The vehicle making the route r starts with a load of $\sum_{i \in S_r} d_i T_r$ units, so the cycle time T_r must also be less than or equal to $\frac{\kappa}{\sum_{i \in S_r} d_i}$. Thus, due to these capacity limitations, the cycle time of route r cannot be higher than the maximum cycle time $T_{max,\,r}$.

$$T_{max,r} = \left\lfloor \min\left(\frac{\kappa}{\sum_{i \in S_r} d_i},\ \min_{i \in S_r} \frac{\kappa_i}{d_i} \right) \right\rfloor \tag{1}$$

One execution of the route takes a certain time t_r and cost C_r for loading and dispatching the vehicle at the depot, driving the actual route and making the deliveries at the customers. If the route is made once every T_r days, this corresponds to a daily cost rate of C_r/T_r.

The average inventory level at customer i is half of its delivery quantity, so $d_i T_r/2$ units. With the inventory holding cost rate of η_i (cost per day per unit), the total stock holding cost at all customers in S_r is then $\frac{T_r}{2} \sum_{i \in S_r} \eta_i d_i$ per day.

When designing a route, the required fleet size is not yet known. However, to take into account the fact that vehicles are eventually needed for making a route and that there is a cost associated to using a vehicle, a vehicle utilization cost term is included in the route cost rate. Specifically, this cost term is $\frac{t_r}{T_r} \cdot \psi$, where t_r is the time it takes to complete route r such that $\frac{t_r}{T_r}$ is the vehicle utilization, i.e., the fraction of time a vehicle is busy making route r. ψ is the fixed cost per day of a vehicle.

The total cost rate TCR_r of the route r replenishing the set of customers S_r is then as follows.

$$TCR_r = \frac{C_r}{T_r} + \frac{T_r}{2} \sum_{i \in S_r} \eta_i d_i + \frac{t_r}{T_r} \cdot \psi \tag{2}$$

This cost rate varies with the cycle time T_r and there is thus an optimal cycle time, denoted T_r^*, for which the cost rate is minimal. This occurs when holding costs at the customers are balanced with the route and vehicle utilization costs.

$$T_r^* = \min\left(\sqrt{\frac{2(C_r + t_r \cdot \psi)}{\sum_{i \in S_r} \eta_i d_i}},\ T_{max,r} \right) \tag{3}$$

For the design of the routes, i.e., the partitioning of the set of customers into subsets each covered by a separate route, a two-phase heuristic was developed. The first phase constructs an initial solution using a savings-based heuristic, and the second phase improves that solution using local search operators. Fig. 1 gives an overview of how the two-phase route design heuristic works.

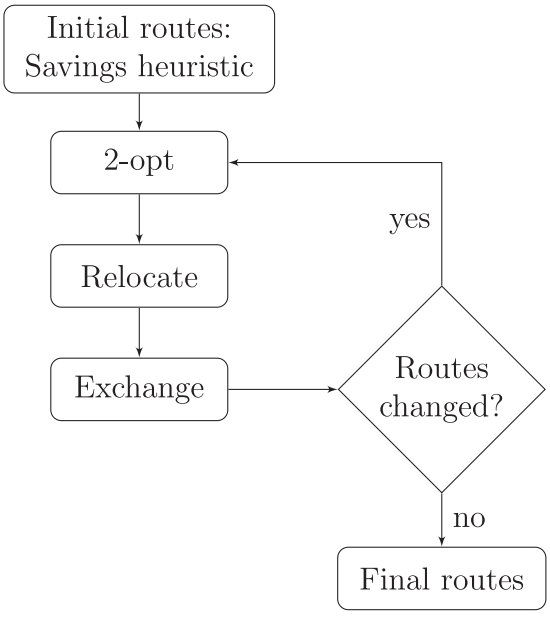

Fig. 1. Flowchart of the two-phase route design heuristic.

The savings-based construction heuristic is an adaptation of the well-known Clarke-and-Wright heuristic (Clarke & Wright, 1964) to the situation of the cyclic IRP. For standard vehicle routing, the evaluation of a merge of two routes in this heuristic consists of checking the vehicle capacity constraint and calculating the distance saving. For the cyclic IRP, however, the evaluation of a merge is more complicated. The vehicle capacity still has to be checked, and the distance saving still has to be determined, but it does not end there. Namely, the merged route's maximal and optimal cycle time have to be calculated in order to minimize the route's cost rate, as explained above.

The second phase of the route design is a local search based improvement phase. The local search procedure can be considered as a variable neighborhood descent (VND) algorithm, consisting of various local search operators that are also well-known from the standard vehicle routing literature. Just like in the construction heuristic of the first phase, however, the evaluation of each of these local search moves had to be extended to the case of the cyclic IRP. Again, it is not only the increase or decrease in distance or travel cost that needs to be evaluated along with feasibility, but the new cost rate of the entire route has to be determined, involving a recalculation of the cycle times and revising the trade-off between distribution and holding costs.

The following standard local search operators (all with quadratic complexity) are being used.

1. 2-opt: remove two arcs (either from the same route or from two different routes) and replace them by two other arcs, such that all routes are closed again and no subtours are created.
2. Relocate: remove a customer and try inserting it into another position (either in the same route, in a different route or in a separate new route).
3. Exchange: switch the positions of two customers (either from the same route or from two different routes).

As shown in Fig. 1, the same local search operator is reiterated until no more improving moves are found before moving on to the next local search operator. Thus, we only consider a single

operator at a time and cycle among the operators until neither finds an improvement anymore.

Further, we adopted a modified best-accept strategy per operator. A single iteration of the local search operators proceeds as follows. All 'new' moves are evaluated and added to a list if they are feasible and improving. A move is 'new' only if it involves nodes that were affected in the previous iteration. (Obviously, in the first iteration of an operator, all moves are new.) Moves that involve nodes that were not affected in the previous iteration, do not need to be reevaluated because their outcome will still be the same as before. By keeping track of affected moves, many move reevaluations are avoided, leading to a much better computational efficiency. The list of new, feasible and improving moves is sorted according to improvement value. After this, all nodes are marked as unaffected and the execution of moves from the list starts. All moves are considered one by one. When a move is encountered that does not involve affected nodes, that move is executed and the nodes involved in the move are marked as affected (to avoid performing two moves with the same nodes in a single iteration). Thus, during a single iteration of an operator, not just the best move is executed, but multiple moves can be executed. This drastically reduces the number of iterations (and corresponding move evaluation and sorting efforts) before the operator runs out of improvements, further improving computational efficiency.

3.2. Fleet design

When a set of cyclic routes is given, the overall solution is also cyclic, with a cycle time equal to the least common multiple of all individual route cycle times. To limit this least common multiple, route cycle times in the route design are limited to $T = 120$ days and all its divisors, i.e., the set {1, 2, 3, 4, 5, 6, 8, 10, 12, 15, 20, 24, 30, 40, 60, 120}. In doing so, the infinite horizon is reduced to a horizon of 120 days and the same solution is reiterated every 120 days.

The second step in building a solution for the cyclic IRP is to build a schedule (with a time horizon of at most 120 days) that indicates which vehicle makes which route on any given day, such that no vehicle has to work more than H hours on any day (e.g., 8 hours), and such that the sum of route cost rates and fixed vehicle fleet cost rates is minimized.

During this fleet design phase, better solutions can often be obtained if the individual route cycle times are modified, as motivated by the following simple example. Consider a solution with only two routes, both taking a full day to complete, with individual optimal cycle times of 2, respectively 3 days. When sticking to these cycle times, both routes have to made together every sixth day, hence two vehicles are needed in the solution. However, when modifying the cycle time from 3 days to 2 days for the second route, only a single vehicle is needed that iteratively makes both tours. As such, aligning route cycle times can help reducing the required fleet size. However, deviating from the optimal route cycle times will increase the individual route cost rates and is therefore only justified if that cost rate increase does not exceed the saving of reducing the fleet. Thus, in this phase, the composition of the routes is not changed, but their cycle times can still be adjusted to maximize vehicle utilization (and minimize the required fleet).

If a route r is made on day t, then it is also made on days $t + T_r$, $t + 2T_r$, etc. Thus, per route, the cycle time T_r has to be selected along with a day t in the first T_r days of the schedule. Then, for all days $t + kT_r, \forall k \in \{0, 1, \ldots, \frac{T}{T_r} - 1\}$, a vehicle must be selected that has enough time left to make the route on that day.

For this fleet design subproblem, the heuristic algorithm of Raa (2015) is adopted. That heuristic also consists of two phases: a construction phase and an improvement phase.

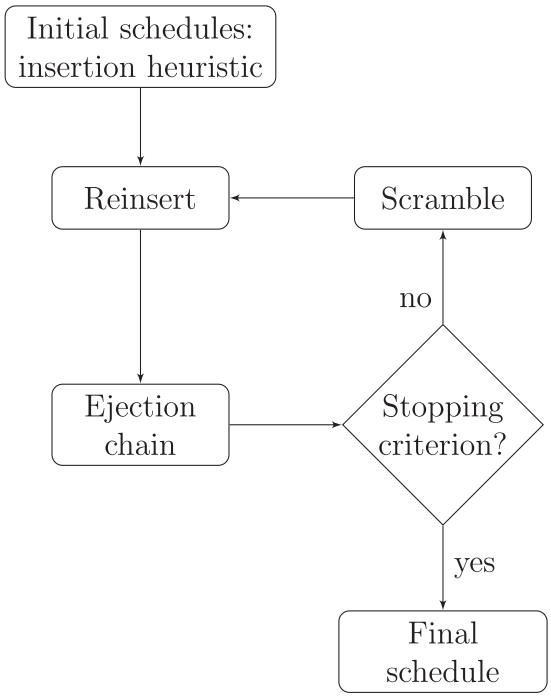

Fig. 2. Flowchart of the two-phase fleet design heuristic.

The construction heuristic is a best-fit insertion heuristic. Each route is inserted into the schedule such that the cumulative remaining time of the vehicles to which the route is assigned on the different days, is minimal. This reflects the idea of keeping vehicle idle time together in bigger blocks, preferably full days. In the construction heuristic, two cycle time selection rules are used, one where routes are inserted with cycle times as close as possible to their optimal cycle times (as this minimizes route cost rates), and one where routes are inserted with cycle times as close as possible to their maximal cycle times (as this minimizes vehicle utilization and hence the fleet cost rate). This leads to two initial schedules that are subsequently passed on to the improvement step.

This improvement step consists of two local-search operators.

1. Remove any single route from the schedule and then reinsert it in the cheapest possible way.
2. Remove all routes being made by the vehicle with the lowest utilization from the schedule and then reinsert them in the cheapest possible way.

To escape the local optimum that is obtained after applying these local search operators, the schedule is *scrambled* by shuffling route allocations among the vehicles according to a random permutation for every day in the schedule horizon. After scrambling, local search is repeated. The stopping criterion for the overall solution approach is a predefined number of scrambles.

Fig. 2 gives an overview of how the fleet design heuristic works. For further details, we refer to Raa (2015).

3.3. Metaheuristic framework

The route and fleet design heuristics explained above build a single solution for the cyclic IRP. With some small adjustments, these heuristics can be reused as the building blocks in a metaheuristic framework that generates multiple solutions (and retains the best found).

The first adjustment is made in the savings heuristic of the route design. When two routes p and q are being merged into

Algorithm 1: Pseudo-code for the evolution strategy.

```
generate initial solution;
while stopping criterion not satisfied do
    m = random number ∈ {1, ..., |S|};
    for i = 1 → m do
        k = random customer from S;
        remove customer k;
    end
    apply route and fleet design heuristics;
end
```

a single route r, the savings s_{pq} according to which the possible merges are sorted, is calculated as follows: $s_{pq} = TCR_p + TCR_q - \boldsymbol{\lambda} \cdot TCR_r$. Varying the parameter λ results in a different order in which routes are being merged and hence to different solutions. Preliminary testing revealed that randomly drawing λ between 0.8 and 1.5 yields relatively good initial solutions that are sufficiently diverse. Values of λ beyond this interval usually result in low-quality solutions.

A second adjustment that is introduced to guide the route design towards different solutions is in the calculation of the route cost rates TCR_r. There, the vehicle utilization cost term is parameterized.

$$TCR_r = \frac{C_r}{T_r} + \frac{T_r}{2} \sum_{i \in S_r} \eta_i d_i + \boldsymbol{\alpha} \frac{t_r}{T_r} \cdot \psi \qquad (2')$$

The parameter α reflects the relative importance of the vehicle utilization cost versus the actual route costs. Varying the parameter α results in different values for the savings and the local search moves and hence also in different solutions. This parameter α will also be randomized, with its value ranging between 0 and 2.

A first framework in which the parameterized route and fleet design heuristics can be embedded, is a basic multi-start heuristic that merely creates new solutions from scratch in every iteration by randomizing the parameters λ and α.

A second solution framework that we propose is a ruin-and-recreate heuristic that iteratively destroys a given solution and then rebuilds it. Ruining (or destroying) a solution is done by removing a random number of customers from the solution, while recreating (or rebuilding) the solution is done by applying the route and fleet design heuristics to the partial solution. This ruin-and-recreate heuristic can be seen as a disruptive mutation operator that is iteratively applied to an individual solution and can hence be considered as an 'evolution strategies' metaheuristic. The pseudo-code for this evolution strategy is given in Algorithm 1.

The third and final framework that we propose is a more elaborate population-based metaheuristic in which the ruin-and-recreate concept is also adopted. Since this metaheuristic combines elements of Genetic Algorithms (GAs) with a Local Search phase, we will refer to it as a 'memetic algorithm'. Although memetic algorithms (MAs) have often been successfully used to address VRPs and VRPTWs, applications to other routing problems such as period routing and inventory routing are less common.

The novelty of the MA that we propose lies in the fact that (i) it only uses the basic building blocks for route and fleet design outlined above, (ii) it does not require complicated chromosome encodings. The crossover operator being applied in the MA also works in a ruin-and-recreate manner. An offspring solution is generated by removing a random number of customers from one of the parent solutions, but only customers that are not incident to common arcs in both parents can be removed. The resulting partial solution after removing some customers is again reoptimized using the route and fleet design heuristics outlined above. The pseudo-code for this crossover operator is given in Algorithm 2.

Algorithm 2: Pseudo-code for the crossover operator.

```
select two parent solutions;
mark customers at the edges of the parents' common arcs;
offspring = copy of first parent;
m = random number ∈ {1, ..., |S|};
for i = 1 → m do
    k = random customer from S;
    if k is unmarked then
        remove customer k from offspring;
    end
end
apply route and fleet design heuristics to offspring;
```

Algorithm 3: Pseudo-code for the memetic algorithm.

```
Data: population size P, stopping criterion S, elite number E
generate P initial solutions using the multi-start heuristic;
while S not satisfied do
    for i = 1 → P do
        create offspring using crossover;
        mutate offspring if it is not unique;
    end
    keep E best solutions in the population;
    calculate diversity scores of all offspring;
    add P − E most diverse offspring to the population;
end
```

Apart from the crossover operator, another important design choice for an MA is population management. Since all of the solutions in the population are being optimized using the local search operators during the route and fleet design, all individuals in the population are local optima. The intensification of the search process is thus already very strong. Therefore, the population management aspect is used to take care of the diversification of the search process. This is done by explicitly measuring the diversity among the individuals in the population. The diversity of two individuals depends on the number of common arcs C and the number of unique arcs U across both solutions and is given by $1 - \frac{C}{C+U}$. Thus, more common arcs result in lower diversity. The overall diversity score of an individual is its cumulative diversity across all individuals in the population. Selecting individuals for crossover is done with the well-known roulette wheel mechanism based on these diversity scores instead of (scaled) objective values. Further, the diversity scores are also used in composing the next generation of the population. This next generation is composed of the best solutions from the population (so-called 'elitism') and the most diverse among the offspring.

Algorithm 3 shows the pseudo-code for the population-based metaheuristic.

For computational testing, the size of the population was set to $P = 30$ solutions after preliminary testing. Per generation, 30 crossovers are performed. To safeguard diversity, only unique individuals are allowed. Therefore, mutation is applied to an offspring if it is the same as a solution already in the population or the set of offspring. From the 60 solutions (30 from the old population and 30 offspring), the $E = 10$ best solutions plus the 20 most diverse offspring constitute the next generation of the population. In the computational experiments below, different stopping criteria are suggested and evaluated.

Table 1
Comparison of multi-start, evolution strategy and memetic algorithm.

	Method	Total cost rate (euro/day)	Fleet (euro/day)	Distr (euro/day)	Inv (euro/day)	CPU (seconds)
S	MS	2020.78	909.63	899.63	211.53	30.03
	ES	2004.31	898.00	895.69	210.62	30.03
	MA	1993.19	895.25	892.18	205.76	29.95
L	MS	3881.21	1609.13	1794.16	477.92	60.08
	ES	3875.94	1612.75	1797.08	466.11	60.07
	MA	3836.79	1603.88	1781.36	451.55	60.22

4. Computational experiments

This section illustrates the performance of the solution approach presented above by applying it to a set of benchmark instances found in the literature. The results show the power of the memetic framework, and the excellent performance of our solution approach compared to other published results for the same benchmarks.

The set of benchmark problems was introduced by Raa (2006) in a 10×2^5 full factorial design with five factors that each have two possible levels and ten instances per factor combination, resulting in a total of 320 instances. The characteristics of the instances are as follows (Raa, 2006).

- The vehicle capacity (VC) is 'Large' (100 units) or 'Small' (50 units). The corresponding fixed costs are $\psi = 400$ and $\psi = 240$ euro per day, and the corresponding variable costs are 1.2 and 1 euro per kilometer. Both large and small vehicles are assumed to have an average speed of 50 kilometer/hour.
- Customer capacity restrictions (CC). When the customer capacity restriction is active (level 'Yes'), the customer storage capacity is generated randomly such that it can hold between 2 to 10 days of supply. Otherwise (level 'No'), the customer storage capacity equals the vehicle capacity.
- The holding cost rate (HC) is either 'High' or 'Low', at $\eta_j = 80$ or $\eta_j = 8$ eurocent per unit per day, $\forall j \in S$.
- Number of customers (NC). The two levels of this factor are $[30 - 70]$ and $[80 - 120]$. This factor is a randomized factor, meaning that instead of considering two fixed values for the factor, values are generated randomly in two distinct intervals.
- Size of the service area (AREA). All customers are located within a circle that has a certain radius and is centered at the depot. The radius of this circle is also a randomized factor, with two intervals [75, 100] kilometer and [150, 175] kilometer.
- Customer demand rates are randomly generated between 1 and 10 units per day. Loading and dispatching the vehicles is assumed to take half an hour ($t_0 = 0.5$hours) and cost 20 euro ($\varphi_0 = 20$ euro), while deliveries at the customers are assumed to take 15 minutes and cost 10 euro ($t_j = 0.25$ hours, $\varphi_j = 10$ euro, $\forall j \in S$).

Results for the 80 instances with large vehicle capacity and large service area are reported in Raa and Aghezzaf (2009), but results for the other 240 instances are available in Raa (2006). Chitsaz, Divsalar, and Vansteenwegen (2016) also report results for all 320 instances.

4.1. Power of the memetic framework

In Section 3, the two-phase method for route design, followed by the two-phase method for fleet design are explained, as well as the metaheuristic frameworks in which these are deployed: the multi-start heuristic (denoted MS), the evolution strategy (denoted ES), and the memetic algorithm (denoted MA). In a first experiment, each of the three solution frameworks is applied to all 320 benchmarks four times, with a runtime of 30 seconds for the small instances and 60 seconds for the large instances as the stopping criterion. The average results across all four runs are shown in Table 1. Results for instances with small and large sets of customers (factor NC) are reported separately.

It can be seen that applying the evolution strategy already improves solution quality compared to the multi-start approach. Across all instances, the average daily cost rate decreases from 2951.00 to 2940.13 euro. This is an improvement of 0.37percent, and the improvement is statistically significant ($t = 6.8$, $p = 1.3e-11$). However, the memetic algorithm improves upon the evolution strategy with another 0.85percent, reducing overall average daily cost rate to 2914.99 euro. Again, this improvement is statistically significant ($t = 16.4$, $p < 2.2e-16$).

In fact, the memetic algorithm manages to substantially improve on all three cost components: distribution costs, inventory holding costs, and fleet costs. Therefore, we can conclude that the combination of favoring diversity among the solutions in the pool and maintaining the common parts during crossover does indeed offer significant added value to the solution framework as it helps find better solutions in the same amount of time.

4.2. Comparison to previous results in the literature

In the second experiment, the results of the memetic algorithm are compared to those obtained by two alternative solution approaches in terms of solution quality and calculation times.

The first alternative is the multi-start heuristic suggested in Raa and Aghezzaf (2009), labeled "CIRH" (Cyclic Inventory Routing Heuristic). In that method, route frequencies for all routes made by the same vehicle are always optimized together, using relative frequencies. As mentioned above, Raa and Aghezzaf (2009) only report solutions for 80 of the 320 instances, but the full results are available in Raa (2006). For our experiments here, we also created a new implementation that is computationally more efficient and in which absolute instead of relative route frequencies are used within the distribution patterns. This new implementation is labeled "CIRH".

The second alternative for which previous results are available is the iterated local search heuristic proposed by Chitsaz et al. (2016), labeled "ILS".

For the newly proposed memetic algorithm, three different versions were each run four times on all instances. The difference is in the stopping criterion. In the first version, "MA1", the runtime is limited to 30 seconds for the small instances and to 60 seconds for the large instances. This corresponds to the MA results from Table 1. In the second version, "MA2", the running time (in seconds) is limited to two times the number of nodes in the instance. E.g., for an instance with 57 customers, the MA is stopped after 114 seconds. Finally, in the third version, "MA3", the stopping criterion is that the MA stops after 1000 generations, or after 500 generations without a new best solution.

Table 2 gives the average total cost rate and the average CPU-time for the single run of the alternative approaches and for the

Table 2
Comparison to other published results.

NC	Method	Total cost rate (euro/day)	CPU (seconds)
S	CIRH (Raa, 2006)	2052.64	232.07
	CIRH'	2054.15	25.81
	ILS (Chitsaz et al., 2016)	2019.50	27.87
	MA1	1993.19	29.95
	MA2	1986.24	85.20
	MA3	1986.31	134.78
L	CIRH (Raa, 2006)	4064.72	2325.89
	CIRH'	4008.53	237.26
	ILS (Chitsaz et al., 2016)	3892.41	151.10
	MA1	3836.79	60.22
	MA2	3827.67	203.98
	MA3	3823.37	385.75

four runs of the different MA versions for the small (S) and large (L) problem instances.

From Table 2, it can clearly be seen that the solution framework we present in this paper outperforms the other methods. In fact, the naive multi-start framework (Table 1) already leads to better results than the best of the alternatives, ILS (2951.00 vs. 2955.96).

On average, MA1 already performs 1.39percent better than ILS, and it does so in about half the CPU-time. When the MA is given more CPU-time, it manages to improve solution quality. However, as can be expected, this improvement slowly tapers off. The improvement from MA1 to MA2 is more substantial than the further improvement from MA2 to MA3. Out of the 320 benchmark instances, MA manages to find no less than 270 new best solutions.

In Table 3, a comparison between CIRH', ILS and MA2 is presented per factor. Note that in this table, the results being reported for MA2 are the best among the four runs instead of the average as in Table 2. These detailed results reveal that MA consistently outperforms ILS, but that CIRH' is capable of occasionally finding better solutions by using one vehicle less. This shows that, although the average performance of MA is much better, there is still some room for improvement in terms of fleet reduction for some of the instances.

5. Conclusion

This paper presents a new solution approach for constructing a cyclic distribution plan for replenishing customers with constant demand rates from a single depot. This solution approach is based on a decomposed view of first creating cyclic routes and then creating vehicle schedules to make the set of routes. For these two phases of route design and fleet design, construction and improvement heuristic building blocks are developed that are embedded in several metaheuristic frameworks. The resulting overall solution approach satisfies the three objectives we set out to achieve (Vidal et al., 2013): (i) it is flexible; (ii) it has a sufficiently simple search guidance strategy; and (iii) it is computationally efficient. This is confirmed by computational testing. The resulting solutions are substantially better than previous results published in the literature, with 270 new best solutions on a set of 320 benchmark instances.

Although the solution approach presented here can be considered state-of-the-art for the cyclic IRP, there are still opportunities for further improvements. A first opportunity is to provide a feedback loop between the fleet design and route design phase. In the current approach, only route cycle times can be adjusted during fleet design. However, if route cycle times have to be changed for reducing the fleet, the distribution vs. holding cost balance of the

Table 3
Detailed benchmarking per factor combination.

					Total cost rate			Improvement		Fleet size		
VC	CC	HC	NC	AREA	CIRH'	ILS	MA2	MA2 vs. CIRH' (percent)	MA2 vs. ILS (percent)	CIRH'	ILS	MA2
L	N	H	L	L	4651.11	4486	4415.15	5.07	1.58	4.6	4.3	4.3
L	N	H	L	S	3299.97	3120.07	3058.42	7.32	1.98	3.4	2.9	2.9
L	N	H	S	L	2297.93	2230.16	2213.86	3.66	0.73	2.5	2.4	2.4
L	N	H	S	S	1784.04	1721.12	1681.03	5.77	2.33	2.0	1.8	1.8
L	N	L	L	L	3072.12	3125.53	3059.37	0.41	2.12	3.8	3.9	3.8
L	N	L	L	S	1966.47	2057.98	1915.12	2.61	6.94	2.4	2.6	2.3
L	N	L	S	L	1578.68	1518.90	1462.87	7.34	3.69	2.2	2.0	1.9
L	N	L	S	S	1057.66	1072.18	1056.36	0.12	1.48	1.4	1.4	1.4
L	Y	H	L	L	6026.99	5709.70	5605.53	6.99	1.82	6.7	6.3	6.3
L	Y	H	L	S	3774.63	3635.17	3562.77	5.61	1.99	4.3	4.1	4.1
L	Y	H	S	L	3106.98	3039.90	2974.17	4.27	2.16	3.7	3.6	3.5
L	Y	H	S	S	2043.00	2022.00	1984.99	2.84	1.83	2.4	2.4	2.4
L	Y	L	L	L	5347.90	4972.57	4956.13	7.33	0.33	6.7	6.2	6.3
L	Y	L	L	S	3108.25	3030.89	2952.84	5.00	2.58	4.1	4.1	4.0
L	Y	L	S	L	2810.23	2734.35	2679.90	4.64	1.99	3.7	3.6	3.5
L	Y	L	S	S	1751.20	1721.09	1706.18	2.57	0.87	2.4	2.4	2.4
S	N	H	L	L	5189.72	5079.75	5026.95	3.14	1.04	7.2	7.3	7.2
S	N	H	L	S	3537.74	3414.26	3346.70	5.40	1.98	5.0	4.8	4.5
S	N	H	S	L	2447.65	2434.67	2397.19	2.06	1.54	3.7	3.8	3.7
S	N	H	S	S	1819.77	1792.63	1774.24	2.50	1.03	2.7	2.7	2.7
S	N	L	L	L	3907.88	3969.98	3923.62	−0.40	1.17	6.7	6.8	6.7
S	N	L	L	S	2437.32	2473.01	2449.40	−0.50	0.95	4.2	4.2	4.2
S	N	L	S	L	1879.73	1913.52	1843.18	1.94	3.68	3.5	3.5	3.3
S	N	L	S	S	1261.33	1315.23	1262.03	−0.06	4.05	2.3	2.4	2.3
S	Y	H	L	L	5849.61	5634.33	5502.94	5.93	2.33	8.8	8.6	8.5
S	Y	H	L	S	3778.76	3629.35	3550.18	6.05	2.18	5.6	5.4	5.3
S	Y	H	S	L	2844.16	2743.03	2723.42	4.25	0.72	4.6	4.4	4.4
S	Y	H	S	S	2000.60	1934.28	1887.03	5.68	2.44	3.2	3.1	3.0
S	Y	L	L	L	5078.32	4917.26	4783.69	5.80	2.72	8.8	8.4	8.2
S	Y	L	L	S	3109.71	3022.72	2953.20	5.03	2.30	5.4	5.3	5.2
S	Y	L	S	L	2514.67	2459.69	2432.26	3.28	1.12	4.5	4.4	4.4
S	Y	L	S	S	1668.84	1659.24	1603.07	3.94	3.38	3.1	3.1	3.0

routes is disturbed. It may then be worthwhile reallocating some customers among the routes to restore that balance, and therefore to change the routes themselves during fleet design instead of only the cycle times. Also, the duration of the different routes could be coordinated during route design, in order to make it easier to combine them in a vehicle schedule afterwards during fleet design.

Another promising avenue for finding better cyclic IRP solutions is to consider so-called 'nested routes' in which not every customer is visited in every iteration of a route. Allowing some customers to be visited only every second or third time a route is made allows for much more opportunities to balance distribution and inventory holding costs, while at the same time it could be reducing vehicle utilization. Of course, this flexibility adds another dimension to the solution space and will therefore make it more complicated to find high-quality solutions.

References

Aghezzaf, E.-H., Raa, B., & Landeghem, H. V. (2006). Modeling inventory routing problems in supply chains of high consumption products. *European Journal of Operational Research, 169*(3), 1048–1063.

Andersson, H., Hoff, A., Christiansen, M., Hasle, G., & Lokketangen, A. (2010). Industrial aspects and literature survey: Combined inventory management and routing. *Computers and Operations Research, 37*(9), 1515–1536.

Anily, S., & Bramel, J. (2004). An asymptotic 98.5%-effective lower bound on fixed partition policies for the inventory-routing problem. *Discrete Applied Mathematics, 145*(1), 22–39.

Anily, S., & Federgruen, A. (1990). One warehouse multiple retailer systems with vehicle routing costs. *Management Science, 36(1)*, 92–114.

Campbell, A. M., & Hardin, J. (2005). Vehicle minimization for periodic deliveries. *European Journal of Operational Research, 165*, 668–684.

Chan, L. M. A., Speranza, M., & Bertazzi, L. (2013). Asymptotic analysis of periodic policies for the inventory routing problem. *Naval Research Logistics, 60*(7), 525–540.

Chitsaz, M., Divsalar, A., & Vansteenwegen, P. (2016). A two-phase algorithm for the cyclic inventory routing problem. *European Journal of Operational Research, 254*(2), 410–426.

Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research, 12*(4), 568–581.

Coelho, L. C., Cordeau, J.-F., & Laporte, G. (2014). Thirty years of inventory routing. *Transportation Science, 48*(1), 1–19.

Larson, R. C. (1988). Transporting sludge to the 106-mile site: An inventory/routing model for fleet sizing and logistics system design. *Transportation Science, 22*(3), 186–198.

Raa, B. (2006). *Models and algorithms for the cyclic inventory routing problem*. Ghent University Ph.D. thesis.

Raa, B. (2015). Fleet optimization for cyclic inventory routing problems. *International Journal of Production Economics, 160*, 172–181.

Raa, B., & Aghezzaf, E.-H. (2008). Designing distribution patterns for long-term inventory routing with constant demand rates. *International Journal of Production Economics, 112*(1), 255–263.

Raa, B., & Aghezzaf, E.-H. (2009). A practical solution approach for the cyclic inventory routing problem. *European Journal of Operational Research, 192*(2), 429–441.

Vidal, T., Crainic, T. G., Gendreau, M., & Prins, C. (2013). Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. *European Journal of Operational Research, 231*(1), 1–21.

Webb, I. R., & Larson, R. C. (1995). Period and phase of customer replenishment: A new approach to the strategic inventory/routing problem. *European Journal of Operational Research, 85*, 132–148.