Customer Consolidated Routing Problem – An Omni-channel Retail Study

Joydeep Paul[a,*], Niels Agatz[a], Remy Spliet[b], René de Koster[a]

[a] *Rotterdam School of Management, Erasmus University, Burgemeester Oudlaan 50, 3062PA Rotterdam, The Netherlands*
[b] *Erasmus School of Economics, Erasmus University Rotterdam, Burgemeester Oudlaan 50, 3062PA Rotterdam, The Netherlands*

Abstract

In this paper, we study a setting in which a carrier can satisfy customer delivery requests directly or outsource them to another carrier. A request can be outsourced to a carrier that is already scheduled to visit the corresponding customer, if capacity allows. For the customers that receive their deliveries directly, we make a vehicle routing schedule that minimizes transportation costs, while for the outsourced customers we incur additional transfer costs between the carriers. This study is motivated by a collaboration with an omni-channel grocery retailer for which goods that are ordered online can be picked up from the stores. The goal is to save costs by consolidating the supply of pick-up points with the store inventory replenishment. To solve this problem, we present exact and heuristic approaches. Computational experiments on both the real-world grocery retail case and artificial instances show that substantial savings can be achieved.

Keywords: Consolidation; Omni-channel retailing; Vehicle routing problem; Local Search

1 Introduction

With the advent of omni-channel retailing, many traditional retailers are now operating online sales channels next to their regular stores. At the same time, pure-play internet retailers are expanding their physical presence by opening up regular stores [1]. An omni-channel service model that is increasingly popular is one that allows customers to buy goods

*Corresponding author
Email address: paul@rsm.nl (Joydeep Paul)

online and then pick them up in a store [2]. According to a recent report [3], 64 percent of Europe's top 500 retailers offer such an in-store pickup service. A similar trend is seen in the U.S.A [4].

There are different fulfillment strategies for this pick-up service model. When the number of pickup orders is small, the goods ordered online can be picked from the store inventory. However, for higher demand volumes, it is often more efficient to pick from a warehouse and then ship to the store. Several large retailers such as Walmart and Tesco [5] [6] use a dedicated warehouse for e-fulfillment tailored to handling B2C orders. This paper focusses on this setting in which the pickup locations at the stores are supplied from a dedicated warehouse.

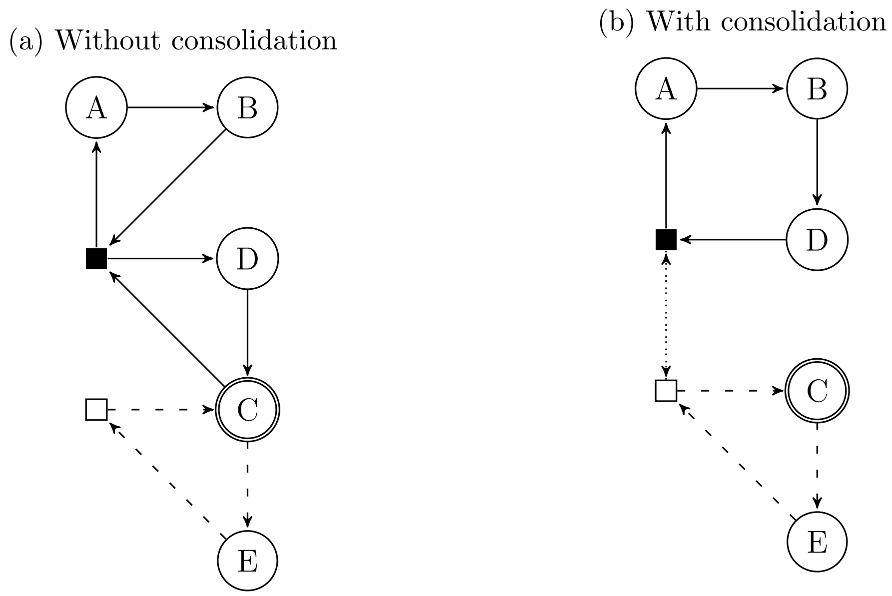
Our research is motivated by a collaboration with the leading omni-channel grocery retailer in the Netherlands. The retailer has grocery stores which also serve as pick-up points (PUP) for goods ordered online. The PUPs are supplied from a dedicated e-fulfillment warehouse, while the store inventory is replenished from one of their traditional warehouses. This means that the same stores are currently visited by different vehicles - one for the replenishment of store inventory and one for the supply of the PUP.

Theoretically, it would be beneficial to jointly plan the supply of the pickup points and the replenishment of the stores. However, this is difficult in practice as none of the involved carriers wants to give up autonomy to a central system. Moreover, various operational constraints limit the flexibility of a possible joint planning. For example, while the replenishment routes need to be planned days in advance to facilitate efficient warehouse operations, the planning of the routes to supply the PUPs can only take place much later due to their short customer lead-times. Therefore, we focus on a new and simple collaboration mechanism in which the replenishment routes are fixed in advance and the PUP supply operations can piggyback on those routes to delivery goods to a set of *shared customers*.

This works as follows. The carrier that fixes its routes first (the *fixed* carrier) communicates the available capacity on its routes to the carrier that plans its routes later (the *flexible* carrier). The flexible carrier can now use that capacity to *outsource* the deliveries of some shared customers to the fixed carrier to reduce the transportation costs and also the number of store visits. To make use of this opportunity, the flexible carrier, however, has to transfer the demands of the outsourced customers to the warehouse of the fixed carrier. For the consolidation to be beneficial, the transfer costs should be less than the savings in the transportation costs.

In Figure 1, we illustrate this consolidation opportunity through an example. When there is no consolidation, the flexible carrier needs two vehicles to serve its four customers $(A, B, C,$ and $D)$ as shown in Figure 1a. Customer C is also served by the fixed carrier. The available excess capacity of the fixed carrier makes it possible for the flexible carrier to outsource the shared customer C to the fixed carrier. Figure 1b shows that as a result of this consolidation, the flexible carrier now only needs to visit three customers, reducing both the travel distances and the number of customer visits. To transfer the demand of the outsourced customers from the warehouse of the flexible carrier to that of the fixed carrier, there is a transfer trip back and forth.

Figure 1: Consolidation of shared customers with a fixed carrier



■ : flexible carrier warehouse, □ : fixed carrier warehouse, ○ : customers, ◎ : shared customer ,
$\rightarrow$: flexible carrier routes, $\dashrightarrow$: fixed carrier routes , $\leftarrow\rightarrow$: transfer trip

In this paper, we introduce the Customer Consolidated Routing Problem (CCRP) which aims to minimize the total costs of the flexible carrier to serve all customers, either directly or by outsourcing to the fixed carrier. As the CCRP reduces to the vehicle routing problem when there is no excess capacity in fixed carrier routes, the CCRP is NP-hard.

Our contribution is threefold. First, we describe a new consolidation strategy with applications in omni-channel retailing. Secondly, we present an exact method and develop several heuristic approaches to solve the associated planning problem. Finally, we present a numerical study to investigate the benefits of the proposed consolidation strategy using both a real-world case and artificial instances.

The remainder of this paper is organized as follows. In the next section, we provide a review of the related literature. In Section 3, we formally describe the problem. Section 4 provides some theoretical properties that are helpful in designing our solution approaches. In Section 5, we present an exact method while in Section 6, we describe several heuristic approaches to solve the problem. Section 7 reports computational results on a real-world case and artificial instances. Finally, Section 8 summarizes our key findings and provides directions for future research.

2 Related literature

The collaboration of logistics carriers is a growing research topic [7, 8]. It is well-known that most efficiency gains can be achieved by jointly planning all logistics operations within a coalition of carriers centrally [7] [9]. However, since carriers typically do not want to give complete autonomy to a central system, they usually plan only part of their operations jointly. Research in this area focusses on the selection of appropriate collaboration partners and mechanisms for exchanging requests among partners [7].

Recent work by Fernández et al. [10] considers the centralized planning in a coalition of carriers in which demands of only a set of shared customers can be transferred between the carriers. In the CCRP, we also consider a set of shared customers between carriers. However, in our setting, one of the carriers fixes its routes in advance while the other can piggyback on those routes.

Conceptually, the CCRP is a selective vehicle routing problem (SVRP) in which only a subset of customers needs to be visited. Most work in this area focuses on settings in which the objective is to maximize the collected profits from the customers given certain constraints on the maximum tour lengths [11]. The SVRP is the multi vehicle version of the selective traveling salesman problem (TSP) [12] or the orienteering problem [13], where a single vehicle visits a subset of customers to maximize the collection of profits from the customers.

A selective vehicle routing problem that is similar to the CCRP is the vehicle routing problem with private fleet and common carrier (VRPPC). In this problem, there is a penalty cost per customer if it is served by an external carrier, and the objective is to minimize the costs to serve all customers either by the private fleet or by an external carrier [14, 15]. Most work on the VRPPC is focussed on the design of heuristics with Tabu search [16, 17] and adaptive variable neighborhood search [18] is currently showing the most promising results.

What distinguishes the CCRP from the existing work in this area is that the number of customers that can be outsourced is constrained by the available capacity in the routes of the fixed carrier. Furthermore, the CCRP explicitly takes into account the transfer trips between the two warehouses as a decision variable.

3 Problem definition

We model the CCRP on a complete directed graph $G = (V, A)$. Here, $V = \{o\} \bigcup N$, where o is the warehouse of the flexible carrier and N is the set of customer locations the flexible carrier has to visit. Each customer $i \in N$ has a demand $q_i \geq 0$, which has to be fulfilled from the warehouse o.

Demand of each customer $i \in N$ can be fulfilled directly by the flexible carrier, or by outsourcing it to the fixed carrier. We do not allow splitting of demand while serving a customer, which means that a customer is visited exactly once by the flexible carrier or its demand is fully outsourced.

To fulfill demand directly, the flexible carrier has a sufficient number of vehicles available, each with capacity Q. For simplicity, we assume $Q \geq q_i, \forall i \in N$. Every vehicle drives a *route*, which is a simple cycle in G starting and ending at the warehouse, and fulfills demand of each customer that is visited along the route. A route is considered feasible if the total demand of the customers that are visited do not exceed the capacity Q. Furthermore, c_{ij} is the cost of traversing an arc $(i, j) \in A$. We assume that c_{ij} satisfies the triangle inequality. Note that c_{ij} might include a service cost for visiting customer $j \in N$.

To fulfill demand by outsourcing, consider the set of *shared customers* $S \subseteq N$ that is visited by both the carriers. Only the demand of these customers can be outsourced. We divide S into disjoint subsets S_r for $r \in R$, where R is referred to as the set of *fixed carrier routes*. As such, S_r can be thought of as the shared customers that are visited by the fixed carrier on a single route. Let $E_r \geq 0$ be the excess capacity available on each fixed carrier route r. For each fixed carrier route r, a set $O_r \subseteq S_r$ is called an *r-outsourcing* if the total demand of the customers in O_r does not exceed the capacity E_r. We define an *outsourcing* as a collection of r-outsourcings $O = \bigcup_{r \in R} O_r$.

The outsourced demand is transported from the warehouse of the flexible carrier to a *transfer point* by means of *transfer vehicles* of capacity Q'. We typically consider the warehouse of the fixed carrier as the transfer point. A fixed cost F is incurred per transfer trip. We assume that a sufficient number of transfer vehicles is available to move the demands of all outsourced customers to the transfer point. It is allowed to split demand of outsourced customers on transfer trips.

The total costs of the flexible carrier comprise of the transfer costs for the outsourced customers and the transportation costs of the customers that are not outsourced. The objective of the CCRP is to determine an outsourcing and corresponding routes for non-outsourced customers so that the total costs are minimized.

The appendix provides a mixed integer linear programming (MILP) formulation for the CCRP based on a two-index formulation for the capacitated VRP [19]. In preliminary experiments, we could solve only very small instances with the straight forward implementation of the MILP using a solver.

4 Theoretical properties

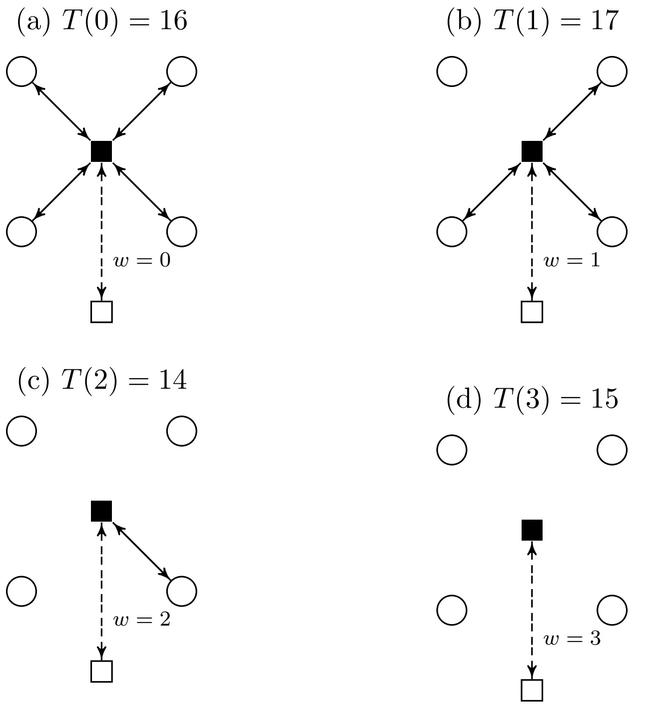
In this section, we present some theoretical properties of the CCRP that help us build our solution strategy. Let w be the number of transfer trips used in a solution and $T(w)$ be the corresponding optimal solution value. The associated optimal routing cost for serving all customers that are not outsourced is given by $R(w)$, hence $T(w) = R(w) + Fw$.

Proposition 1. *The optimal cost, $T(w)$, of CCRP is in general neither convex nor concave in w.*

Proof. We prove this proposition by providing an instance for which $T(w)$ is neither convex nor concave in w. Consider an instance with four customers where each of the customers has a demand of $\frac{4}{7}Q$ and cost of delivering to each of them from the warehouse is 4. The capacities of the vehicles of the flexible carrier and the transfer vehicles are the same, i.e., $Q = Q'$. The transfer cost per trip F is 5. The excess capacity of the fixed carrier routes is such that all the customers can be outsourced. Observe that because demand cannot be split over flexible carrier visits, every non-outsourced customer is visited by a separate vehicle.

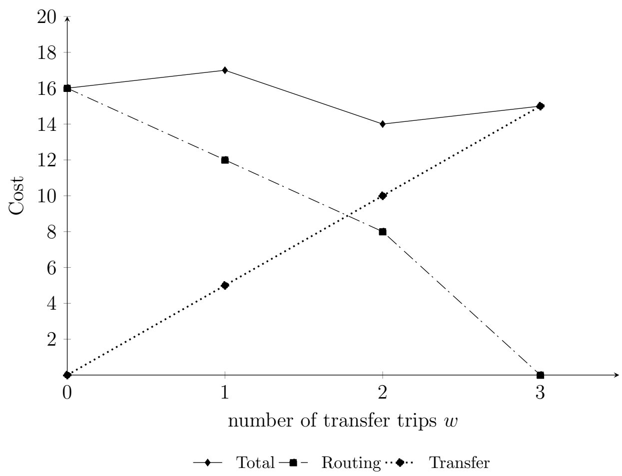
When $w = 0$, no customers are outsourced and every customer is visited by a separate vehicle, hence $T(0) = 16$. For $w = 1$, the optimal decision is to outsource one customer to the fixed carrier, so $T(1) = 17$. In case $w = 2$, three customers can be outsourced, now it follows that $T(2) = 14$. Finally, for $w = 3$, it is optimal is to outsource all four customers, hence $T(3) = 15$. We show the optimal solutions when w is fixed to values 0, 1, 2 and 3 in the Figures 2a, 2b, 2c and 2d respectively. The optimal solution values are plotted in Figure 3. Clearly, $T(w)$ is neither convex nor concave for this instance. $\square$

Figure 2: Optimal solutions for our example ($Q' = Q; q_i = \frac{4}{7}Q; F = 5; c_{oi} = 2, \forall i \in N$)



■ : warehouse, □ : transfer point, ○ : customers, ↔: flexible carrier routes, ←→: transfer trip

Figure 3: Costs of optimal solutions for our example



Since $T(w)$ is in general not convex or concave in w, we pursue an enumerative strategy over w. Next, we show how to bound our search, by defining an upper bound on w.

Proposition 2. *The following are all upper bounds on w^*, the number of transfer trips in the optimal solution of the CCRP.*

- $UB_1 = \left\lceil \frac{\sum\limits_{r \in R} E_r}{Q'} \right\rceil$,
- $UB_2 = \left\lceil \frac{\sum\limits_{i \in N} q_i}{Q'} \right\rceil$,
- $UB_3 = \left\lfloor \frac{R(0) - R(X)}{F} \right\rfloor$,

where X is an upper bound on w^*, for instance $X = \min\{UB_1, UB_2\}$.

Proof. The total excess capacity in the fixed carrier routes is given by $\sum\limits_{r \in R} E_r$. Since we can split demands in transfer trips, the number of transfer trips required to fully utilize the available excess capacity is $\left\lceil \frac{\sum\limits_{r \in R} E_r}{Q'} \right\rceil$. Hence, $w^* \leq \mathrm{UB}_1$. Similarly, the total outsourced demand is limited by $\sum\limits_{i \in N} q_i$, yielding $w^* \leq \mathrm{UB}_2$

Next, we prove that $w^* \leq \mathrm{UB}_3$. As before, denote by $R(w)$ the optimal routing costs when using w transfer trips. Observe that $R(0) \geq R(w^*) + Fw^*$. When the arc costs satisfy the triangle inequality, R is decreasing in w. Therefore, it holds for $X \geq w^*$ that $R(w^*) + Fw^* \geq R(X) + Fw^*$. Combining these observations yields $w^* \leq \left\lfloor \frac{R(0) - R(X)}{F} \right\rfloor$. □

Combining the bounds presented by Proposition 2, we can bound the optimal number of transfer trips w^* by $\mathrm{UB} = \min(\mathrm{UB}_1, \mathrm{UB}_2, \mathrm{UB}_3)$. Note that UB_1 and UB_2 can be computed efficiently, while UB_3 requires solving one VRP ($R(0)$) and one CCRP ($R(X)$) with given number of transfer trips.

Next, for a fixed number of transfer trips w, we limit the set of outsourcings that we consider when searching for an optimal solution. We define an outsourcing to be *maximal* if no additional customers can be outsourced without violating the available transfer capacity wQ' or the total available excess capacity $\sum_{r \in R} E_r$. We can similarly define maximality of an r-outsourcing. Note that not all r-outsourcings that are part of a *maximal* outsourcing are necessarily *maximal* themselves.

Proposition 3. *There exists an optimal solution of the CCRP for which the outsourcing is maximal.*

Proof. Assuming that the triangle inequality holds, we know that the routing cost is decreasing with the number of outsourced customers. Hence, if the outsourcing is not maximal, an additional customer demand can be outsourced without increasing the costs. □

We can now reformulate our problem in the following way:

$$\min_{0 \leq w \leq UB} \{Fw + \min_{O \in O_{max}} R(N \backslash O)\}$$

where, O_{max} is the set of all maximal outsourcings, and $R(S)$ denotes the routing cost for the set of customers S. Next, we present a solution procedure in which we enumerate over all relevant values of w and subsequently solve the subproblem of finding a maximal outsourcing that minimizes the corresponding routing costs.

5 Exact solution approach

To solve the problem to optimality, we enumerate the number of transfer trips w from 0 to UB. For each value of w, we enumerate all maximal outsourcings. Finally, for every maximal outsourcing, we solve the vehicle routing problem visiting the non-outsourced customers. The best found solution is optimal.

For a given maximal outsourcing, the CCRP reduces to a standard capacitated VRP. We use a standard branch-and-cut procedure to solve the VRP, in which we make use of a 2-index flow formulation including the well known rounded capacity constraints [19]. We relax the rounded capacity constraints, identify all violated rounded capacity constraints when a feasible integer solution is found and add these to the formulation.

To further speed up our solution procedure, we keep track of the current best solution to the CCRP as a lower bound to terminate the evaluation of certain outsourcings. If at any stage of the branch-and-cut procedure to solve a vehicle routing problem, the lower bound plus the transfer costs for the incumbent solution is higher than the current best solution, we discontinue the evaluation of this outsourcing and continue with the next.

6 Heuristics

In the exact approach as described in the previous section, we enumerate all maximal outsourcings and solve the associated VRP to find the optimal solution. When the number of customers grows large, the number of maximal outsourcings may also grow large, making enumeration computationally intractable. Moreover, the computation time for solving a VRP to optimality also grows. This makes it intractable to solve a VRP for all maximal outsourcings, and if the number of customers is sufficiently large, it is even intractable for a single maximal outsourcing. Hence, for large instances, we develop heuristics to identify a promising maximal outsourcing and solve the corresponding routing problem.

6.1 Knapsack heuristic

Next, we present a constructive heuristic, which we refer to as the Knapsack heuristic. Instead of evaluating all maximal outsourcings, we try to find promising outsourcings by solving a knapsack problem. We assign a profit c_i for outsourcing customer $i \in N$, and search for an outsourcing that maximizes the total profit. In our experiments, we consider different profits yielding the following objectives:

- *Maximize the sum of the distance to the warehouse of all outsourced customers*
- *Maximize the number of outsourced customers (cardinality)*
- *Maximize sum of the demand of all outsourced customers*

Note that if the size of the demands of all customers are identical, maximizing the sum of the demand of all outsourced customers is same as maximizing the number of outsourced customers.

We can formulate the corresponding optimization problem as a multiple Knapsack problem [20] where each *knapsack* represents an r-outsourcing. Let the variable y_i be 1 if customer i is outsourced to the fixed carrier, and 0 otherwise. The multiple knapsack problem is formulated as follows:

$$
\begin{aligned}
\max \quad & \sum_{i \in N} c_i y_i & & & \\
\text{s.t} \quad \sum_{i \in S_r} q_i y_i & \leq E_r & & \forall r \in R & (1)\\
\sum_{i \in N} q_i y_i & \leq Q' w & & & (2)\\
y_i & \in \{0,1\} & & & (3)
\end{aligned}
$$

The capacity constraints of the fixed carrier routes are captured in constraints (1). Constraint (2) ensures that the total demand of the outsourced customers fit into w transfer vehicles.

As solving a VRP to evaluate the cost of a particular outsourcing is computationally intractable for larger instances, we use our implementation of the adaptive large neighborhood search (ALNS) heuristic by Pisinger and Ropke [21]. This approach uses a local search framework based on simulated annealing and several destroy and repair operators.

For each $w = 1, \ldots, UB$, we determine the promising maximal outsourcing by solving the above multiple knapsack problem, and solve the associated VRP to evaluate the outsourcing. The solution with the least total cost, i.e., routing and transfer costs, is chosen.

6.2 Improvement phase

To further improve the solution obtained from the Knapsack heuristic, we develop an improvement heuristic in which we iterate between an *intensification* phase and a *diversification* phase. In the intensification phase, we improve the solution quality by a neighborhood search procedure. In the diversification phase, we attempt to move away from the local optimum. If the intensification and diversification do not lead to an improvement of the *best* solution for I iterations, we terminate.

Intensification

At each iteration, we search an r-outsourcing exchange neighborhood that is specific to our problem. The search continues until no more improving r-outsourcing exchange is found. For every outsourcing considered during the search, including the initial outsourcing, we use standard 1-point moves and swaps to optimize the corresponding routes. Next, we provide a brief summary of these neighborhoods.

1-point move and swap neighborhood

A 1-point move is a repositioning of a single customer among routes in the solution. Only at initialization of the intensification phase we also consider outsourced customers for repositioning. In that case, we do not only consider repositioning customers somewhere in a route but we also consider outsourcing customers currently included in a route. Similarly, we use swaps to exchange the positions of two customers.

We consider the 1-point move and swap together in a single neighborhood. This means that the best of all possible 1-point moves and swaps across all customers is implemented at each iteration.

r-outsourcing exchange neighborhood

An *r-outsourcing exchange* exchanges an r-outsourcing O_1 in the current solution with another r-outsourcing O_2 for fixed carrier route $r \in R$. We perform this exchange as follows. We remove all customers in O_1 and O_2 from the solution. Next, we outsource the customers in O_2. All the remaining customers are inserted to the flexible carrier routes in random order at the cheapest position. Subsequently, we re-optimize the flexible carrier routes with the 1-point move and swaps until no more improvement is found. The difference in the total cost before and after the exchange gives the improvement of the exchange.

The r-outsourcing exchange corresponding to the best improvement is implemented at each iteration.

Diversification

If no more improving moves can be found, we implement a 'destroy and repair' strategy. In particular, we remove m customers from the solution and insert them back to form a feasible solution. The values for m are generated randomly between an instance defined *lower* and *upper* limits which depend on the parameters $\delta < 1$, $\gamma < 1$, l and u in the following way:

$$lower = minimum\{\delta|V|, l\} \qquad upper = minimum\{\gamma|V|, u\}$$

This *destroy* operation is similar to the *destroy* operation in the ALNS heuristic by Pisinger and Ropke [21].

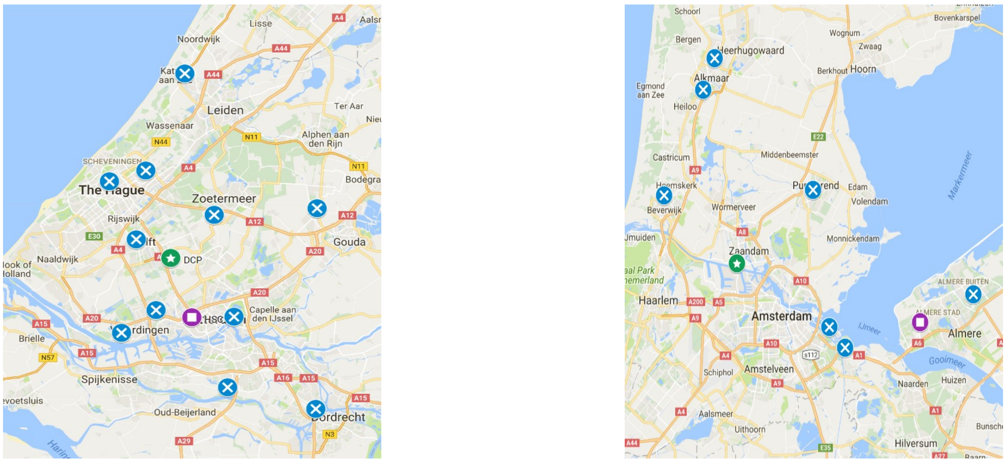
During the repair stage, a customer can either be outsourced to the fixed carrier or served by the flexible carrier. If it is feasible to be outsourced, we assign it to the fixed carrier with a probability ρ, otherwise we insert the customer at the first position of the first route of the flexible carrier with sufficient capacity.

7 Computational study

In this section we report the results of our computational experiments. The goal of these experiments is to assess the quality of our heuristics and the benefits of consolidation with a fixed carrier under different settings. All algorithms are coded in JAVA and Gurobi 7.0 is used as the MILP solver. The experiments were performed on a laptop computer with an Intel Core i7-4810MQ CPU 2.8 GHz processor.

7.1 Real-world case study

To assess the potential savings of our omni-channel consolidation strategy, we apply our model to the transportation network of a large grocery retailer in the Netherlands. Some of the retailer's grocery stores also serve as PUPs for groceries ordered online. To enable efficient order picking, the PUPs are supplied from one of three e-fulfillment warehouses, while the inventory of the same stores are replenished by one of four regional warehouses.



(a) South-West, 11 stores

(b) North-West, 7 stores

Figure 4: Locations of PUPs (cross), Regional (star) and E-fulfillment warehouses (square)

We use route data from ten days in February 2017 for store replenishment and PUP supply for two regions in the Netherlands. Figure 4 shows the locations of the regional warehouses, e-fulfillment warehouses and grocery stores in the two regions. Figure 4a shows the South-West (SW) region where there are eleven stores with a PUP that are served by both carriers. Similarly, Figure 4b shows the seven PUP stores are served by both carriers in the North-West (NW) region. In this case, the store replenishments take place before the time that the in-store pickup points open to the customers so all outsourcings are time feasible.

The demands of the stores and capacities of the vehicles are given in numbers of roll cages. We assume that the transfer cost per trip are proportional to the return distance between the two warehouses. The network structures of the two cases are similar but the transfer costs are significantly different, i.e., 20.8 km for SW and 76 km for NW. To create a benchmark for the case without consolidation, we determine the optimal routing costs by solving a VRP in which the flexible carrier visits all *shared* customers, i.e., the stores that have a PUP.

We assume that the transfer trips are done by the same type of vehicles as the PUP deliveries, i.e., $Q' = Q = 50$ roll cages. Since the number of stores is relatively small, we solve the instances using the exact method described in Section 5. Table 1 shows the results of consolidation for the SW and NW cases. We report the savings in routing cost relative to the routing costs without consolidation while the savings in the number of store visits is relative to number of stores in the instances.

Table 1: Savings by consolidation in real-world case studies ($F = 20.8$ for SW, $F = 76$ for NW)

Instances	Days with consolidation	Savings: Transport cost* (%)	Savings: Store visit* (%)
SW - 11 stores	10 / 10	33.4	60.9
NW - 7 stores	4 / 10	1.6	60.0

*average over days with consolidation

Table 1 shows that consolidation results in average cost savings of 33.4% for the SW case and only 1.6% for the NW case. Moreover, we see that it is beneficial to consolidate in all ten days in the SW case and only in four out of the ten days in the NW case. One important reason for the different savings is the fact that the transfer distance and the associated transfer costs are much higher for the NW case than for the SW case. This means that in the NW case, much of the routing costs savings are offset by the additional transfer costs. In both cases, we do observe a 60% reduction in the number of store visits for the days with consolidation.

While the current instances are small enough to be solved by our exact approach, the retailer wants to convert many more stores into PUPs which would create larger instances. In the next section, we generate larger instances to test our heuristics.

7.2 Generation of artificial instances

We generate artificial instances based on the VRP instances from the VRP-lib [22]. In particular, we use these instances to represent the set of *shared* customers that is served by both the flexible and the fixed carrier. For the flexible carrier we use the customer demand and vehicle capacities as given in these instances and assume the same capacities for the transfer trips. The best known solutions for these instances represent the benchmark solutions for the situation without consolidation.

To generate the fixed carrier routes, we randomly generate demand sizes between one and five for the customers in the instances and then solve a VRP. We create different routes by using different vehicle capacities in the VRP. The parameter c indirectly determines the vehicle capacity which is $\left\lceil c \sum\limits_{i \in N} \frac{q_i}{N} \right\rceil$. This way, external routes with roughly c customers are generated. We then generate the excess capacity E_r of route r by drawing a value between the smallest demand of the customers in the route and e times the total demand of the customers in the route. Note that, if this interval is empty, the excess capacity is randomly generated between the smallest demand of the customers in the route and the total demand of the customers in the route i.e., $e = 100\%$. This ensures that at least one customer can feasibly be outsourced to each fixed carrier route. We set the transfer costs to 0.5 times the maximum distance between two locations in the graph.

7.3 Performance of heuristics

To evaluate the performance of our heuristics, we use the VRP-lib instance of size 32 (including the depot) for which we are able to find optimal solutions by using the approach described in Section 5. We test our heuristics on different instances generated using different values for c and e, i.e., e = 20,30,40% and c = 2.5,5,10. For each combination of c and e, we generate five random instances. To provide more insights into the characteristics of each set of instances, we report the average number of fixed routes (FR) and the average number of possible outsourced customers (OM). To calculate the latter statistic, we determine the maximum number of customers that can be outsourced given the available excess capacity.

We use three performance measures to evaluate the heuristics: average optimality gap, maximum optimality gap and number of times the optimal solution is found.

The three objectives to maximize distance, demand and cardinality are used in the knapsack heuristic. Subsequently, we implement the improvement phase on the solution obtained with each measure. The *DisK, DemK* and *CarK* correspond to the solutions of knapsack heuristics obtained with the objectives *distance, demand* and *cardinality* respectively, while $DisKI$, $DemKI$ and $CarKI$ correspond to the solution values after the improvement phase.

For the heuristic, we use the parameters as in [21] as provided in Table 2.

Table 2: Parameter settings of the heuristics

Parameters	Description	Values
δ	lower bound parameters on the number	0.1
l	of customers to be removed in the destroy phase	30
γ	upper bound parameters on the number	0.4
u	of customers to be removed in the destroy phase	60
ρ	probability of outsourcing a customer in the repair stage	0.5
I	number of iterations of the improvement phase	100

Table 3: Benchmarking of the heuristics ($|V| = 32, Q = Q' = 100$)
(summary of five random realizations)

FR	OM		DisK	DemK	CarK	DisKI	DemKI	CarKI
			Heuristics					
3.8	6.6	Avg Δ (%)	1.05	2.10	2.18	**0.00**	0.02	**0.00**
		Max Δ (%)	4.29	4.68	4.10	**0.00**	0.08	**0.00**
		# Opt	1/5	0/5	0/5	**5/5**	4/5	5/5
3.8	9.0	Avg Δ (%)	2.28	6.02	3.73	**0.00**	0.40	0.05
		Max Δ (%)	5.82	11.15	8.97	**0.00**	1.58	0.19
		# Opt	0/5	0/5	0/5	**4/5**	**4/5**	**4/5**
3.8	12.6	Avg Δ (%)	3.34	4.97	6.21	**0.00**	0.03	**0.00**
		Max Δ (%)	7.66	7.38	12.13	**0.00**	0.15	0.01
		# Opt	0/5	0/5	0/5	**5/5**	4/5	4/5
6.6	8.6	Avg Δ (%)	1.49	1.65	1.87	0.06	**0.00**	**0.00**
		Max Δ (%)	3.78	2.98	3.22	0.32	0.32	**0.00**
		# Opt	0/5	0/5	0/5	4/5	3/5	**5/5**
6.6	9.6	Avg Δ (%)	1.86	1.54	2.26	0.04	**0.00**	0.04
		Max Δ (%)	5.21	3.65	4.69	0.22	**0.00**	0.22
		# Opt	0/5	0/5	1/5	4/5	**5/5**	4/5
6.6	12.0	Avg Δ (%)	1.06	2.00	1.51	**0.06**	0.13	0.14
		Max Δ (%)	2.00	3.99	5.53	**0.30**	0.66	0.69
		# Opt	0/5	0/5	2/5	**4/5**	**4/5**	**4/5**
12.2	14.0	Avg Δ (%)	1.91	6.32	5.06	0.82	**0.50**	0.52
		Max Δ (%)	4.50	10.34	9.36	1.75	**1.32**	1.43
		# Opt	0/5	0/5	0/5	0/5	**3/5**	1/5
12.2	13.2*	Avg Δ (%)	1.49	3.50	4.32	**0.00**	0.08	0.03
		Max Δ (%)	2.59	7.06	7.06	**0.00**	0.33	0.12
		# Opt	0/4	0/4	0/4	**4/4**	3/4	3/4
12.2	12.6	Avg Δ (%)	1.54	3.79	3.93	0.20	0.10	**0.01**
		Max Δ (%)	2.59	6.01	9.10	0.98	0.50	**0.04**
		# Opt	1/5	0/5	0/5	**4/5**	**4/5**	**4/5**
Total		Avg Δ (%)	1.78	3.54	3.45	0.14	0.14	**0.09**
		Max Δ (%)	7.66	11.15	12.13	1.75	1.58	**1.43**
		# Opt	2/44	0/44	3/44	**34/44**	**34/44**	**34/44**

*average of 4 realizations as one of the instances could not be solved to optimality within 24 hours

Table 3 reports the average and maximum optimality gaps and the number of times the optimal solution is obtained out of the five realizations for each combination of c and e. Considering the average across all instances, the optimality gap is less than 4% for the three basic approaches (DisK, DemK, & CarK) with a maximum gap of 12.13% . The DisK and CarK heuristics find the optimal solutions in two and three cases respectively, while the DemK does not find any optimal solution.

The improvement phase applied to the solutions of the knapsack heuristics provide significant gains. The average optimality gap is less than 0.2% for all approaches ($DisKI$, $DemKI$ and $CarKI$) and the maximum optimality gap is less than 2%. Moreover, they find the optimal solution for 77% of the instances. The solution quality for the three approaches is comparable.

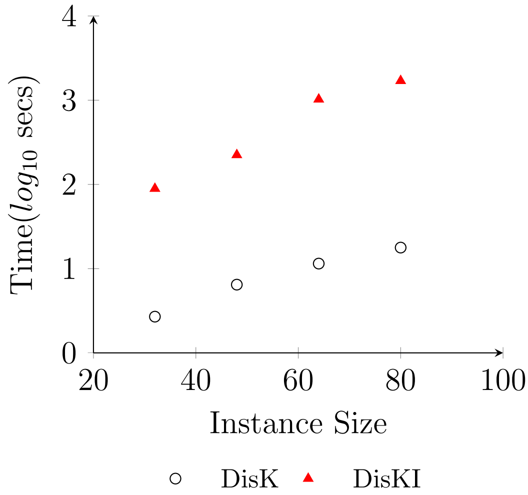
To further evaluate the performance of the different heuristics, we test their performance on larger instances. Since we cannot find the optimal solutions for these instances, we compare to the best solution for each instance among the heuristics. Table 4 shows the results of the experiments for instances of sizes 48-80. DisKI is the best performing heuristic with an average gap of 0.03% and a maximum gap of 0.1% from the best solution. Hence, we use this heuristic to investigate the benefits of consolidation for the instances.

Table 4: Performance of the heuristics (summary of five random realizations)

Instance	FR	OM		Heuristics					
				DisK	DemK	CarK	DisKI	DemKI	CarKI
A-n48-k7	9.8	13.8	Avg Δ (%)	2.59	4.18	4.25	0.00	0.07	0.22
			Max Δ (%)	3.84	7.87	7.08	0.00	0.37	0.78
			# Best	0/5	0/5	0/5	5/5	4/5	3/5
A-n64-k9	12.4	18.4	Avg Δ (%)	1.65	2.50	3.40	0.10	0.06	0.00
			Max Δ (%)	1.88	3.78	4.51	0.29	0.29	0.13
			# Best	0/5	0/5	0/5	1/5	3/5	4/5
A-n80-k10	16.0	26.0	Avg Δ (%)	1.73	1.69	2.91	0.00	0.06	0.39
			Max Δ (%)	3.87	3.19	4.29	0.01	0.19	1.77
			# Best	0/5	0/5	0/5	4/5	3/5	2/5
Total			Avg Δ (%)	1.99	2.79	3.52	**0.03**	0.06	0.20
			Max Δ (%)	3.20	4.95	5.29	**0.10**	0.28	0.89
			# Best	0/15	0/15	0/15	**10/15**	**10/15**	9/15

Figure 5 shows the solution time (in log scale) of the DisK and DisKI heuristics. The DisK performs very fast compared to the DisKI. While there is an improvement in the quality of the solution with the improvement phase, the time required for the improvement increases with the instance size.

Figure 5: Computation times of the DisK & DisKI heuristic



7.4 Savings by consolidation across different instances

Next, we present the results of the experiments with larger instances from the VRP-lib repository using the *DisKI* heuristic. Similar to Section 7.3, we generate the fixed carrier routes for these instances using $e = 20, 30, 40\%$ and $c = 2.5, 5, 10$. The instance descriptions for the scenarios without consolidation are given in Table 5.

Table 5: Description of the instances (No consolidation)

Instance	Transport cost (km)	Transfer cost per trip (km)
A-n32-k5	784	64
A-n48-k7	1,073	60
A-n64-k9	1,401	59
A-n80-k10	1,764	69

Table 6 shows the relative costs savings and the number of customer-visits as compared to the setting without consolidation across all instances. For each instance, we report the average values over five random realizations of the fixed carrier routes.

We observe that within the instances of same number of fixed carrier routes r, the savings in transport costs and customer-visits increase as the average maximum number of outsourced customer increases.

Table 6: Savings (%) by consolidation using DisKI heuristic (summary of five random realizations)

Instance	FR	OM	Savings	
			Transport cost (%)	Customer visit (%)
A-n32-k5	3.8	6.6	2.7	18.7
	3.8	9.0	6.2	24.5
	3.8	12.6	13.2	32.9
	6.6	8.6	1.9	27.1
	6.6	9.6	5.2	28.4
	6.6	12.0	10.5	36.1
	12.2	14.0	12.8	37.4
	12.2	13.2	9.7	38.7
	12.2	12.6	10.4	36.8
A-n48-k7	5.0	9.4	3.7	16.2
	5.0	12.8	11.1	20.4
	5.0	17.6	15.3	26.8
	9.8	12.6	9.4	24.7
	9.8	13.8	10.5	24.7
	9.8	15.0	11.7	23.8
	18.2	20.4	18.1	40.4
	18.2	19.4	14.5	40.4
	18.2	19.6	15.0	40.4
A-n64-k9	7.0	18.0	9.4	23.8
	7.0	19.8	12.4	24.1
	7.0	25.8	17.1	35.2
	12.4	17.0	8.8	22.5
	12.4	18.4	9.8	25.4
	12.4	23.4	14.8	34.0
	23.4	27.8	17.6	39.4
	23.4	25.2	15.7	36.2
	23.4	26.4	16.9	38.1
A-n80-k10	8.0	20.8	9.2	25.6
	8.0	26.2	11.9	25.1
	8.0	31.4	16.8	32.7
	16.0	21.6	9.0	25.1
	16.0	26.0	10.9	30.6
	16.0	28.4	12.9	32.9
	30.8	34.8	16.7	39.7
	30.8	33.2	14.9	39.7
	30.8	33.4	15.0	38.7

7.5 *Impact of the service cost*

The consolidation of deliveries creates transportation cost savings and also reduces the number of stops at the various customer locations. This stop reduction may lead to service costs savings which reflect savings in the time required at the customer location to make a delivery such as parking and (un)loading. In this section, we investigate the impact of the potential service cost savings of customer consolidation on the solutions of the CCRP. Therefore, we introduce a parameter τ that represent the service costs that can potentially be saved by not visiting a customer location by the flexible carrier. We define τ relative to the transfer cost per trip and use the *DisKI* heuristic to obtain the solutions for different values of τ. We use one of the instances of size 48 as described in Section 7.2 for this experiment.

Table 7 presents the transportation cost savings and the service cost savings due to consolidation ('WITH CONSOL') for different values of τ as compared to the situation without consolidation ('NO CONSOL'). Moreover, we report the number of outsourced customers and the number of transfer trips. We compare the costs of the solution of the model that does not explicitly take the service costs into account ('TRANSPORT') and the solution of the model that explicitly takes the service costs into account ('SERVICE'). As explained in Section 3, we can take the service costs into account by adding a costs to each arc that ends at a customer. We normalize all costs by dividing them by the routing cost of the solution without consolidation.

The results show that both solutions are the same for $\tau = 0$ as there are no additional service costs savings associated with consolidation. For $\tau = 20$, we see that the transportation costs slightly increases in the SERVICE solution as compared to the TRANSPORT solution to create more service costs savings. Since the number of transfer trip remains the same, this implies that outsourcing one additional customer (13 instead of 12) results in higher routing costs. This may be counterintuitive as in this case the flexible carrier serves one customer less. To understand this, we should realize that it may be possible to achieve more routing cost savings by outsourcing one customer that is far away as compared to two customers that are close by.

For $\tau = 100$, we see a further increase in the transportation costs and also in the number of outsourced customers. This solution requires an additional transfer trip. However, the cost of this additional transfer trip is offset by the service costs savings. Note that since the number of outsourced customers is constrained by the available excess capacity of the fixed carrier, it can increase only up to a certain limit, which in this case is 14. This means that the solution will not change when we further increase τ.

Table 7: Change in the savings with change in service costs ($|V| = 48$, $Q = Q' = 100$)

τ (%)	NO CONSOL Transport cost	NO CONSOL Service cost	WITH CONSOL Scenario	Transfer trips	Outsourced nodes	Transport cost savings	Service cost savings	Total savings
0	100	0	TRANSPORT	1	12	10.7	0.0	10.7
			SERVICE	1	12	10.7	0.0	10.7
20	100	52	TRANSPORT	1	12	10.7	13.4	24.1
			SERVICE	1	13	9.8	14.5	24.3
100	100	262	TRANSPORT	1	12	10.7	66.9	77.6
			SERVICE	2	14	4.0	78.0	82.0

8 Conclusion

We introduce the customer consolidated routing problem(CCRP) in omni-channel retail distribution. Our consolidation strategy enables one carrier to make use of the excess capacity of another carrier to reduce the total transportation costs and also the number of delivery stops. The computational study on both the real-life case and the artificial instances shows that substantial cost savings can be achieved by consolidating customers. Our heuristics provide good quality solutions in a reasonable time.

A potential future research direction is to develop better exact solution procedures. This will help to benchmark the performance of the heuristics for larger instances as well. Considering the scope of the research, an extension of the problem may consider multiple transfer points instead of a single one. While this could potentially lead to higher savings, this will also increase the complexity of the associated planning problems.

References

[1] G. Speculations, Why Would Amazon Open Physical Stores? (Feb 2016).
URL https://www.forbes.com/sites/greatspeculations/2016/02/11/why-would-amazon-open-physical-stores/#44c11c2f964d

[2] F. Gao, X. Su, Omnichannel retail operations with buy-online-and-pick-up-in-store, Management Science 63 (8) (2016) 2478–2492.

[3] I. Jindal, A performance ranking of Europe's Top500 ecommerce and multichannel retailers, http://viewer.zmags.com/publication/5f09e229#/5f09e229/22 (2017).

[4] P. Rosenblum, B. Kilcourse, Omni-channel 2013: The long road to adoption, RSR 2013 benchmark report, Retail Systems Research (2013).

[5] N. Bose, Wal-Mart's next move against Amazon: More warehouses, faster shipping, https://www.reuters.com/article/us-walmart-ecommerce/wal-marts-next-move-against-amazon-more-warehouses-faster-shipping-idUSKCN12609P (2016).

[6] A. Hübner, H. Kuhn, J. Wollenburg, Last mile fulfilment and distribution in omni-channel grocery retailing: a strategic planning framework, International Journal of Retail & Distribution Management 44 (3) (2016) 228–247.

[7] M. Gansterer, R. F. Hartl, Collaborative vehicle routing: a survey, arXiv preprint arXiv:1706.05254.

[8] F. Cruijssen, M. Cools, W. Dullaert, Horizontal cooperation in logistics: opportunities and impediments, Transportation Research Part E: Logistics and Transportation Review 43 (2) (2007) 129–142.

[9] C. Lin, A cooperative strategy for a vehicle routing problem with pickup and delivery time windows, Computers & Industrial Engineering 55 (4) (2008) 766–782.

[10] E. Fernández, M. Roca-Riu, M. G. Speranza, The shared customer collaboration vehicle routing problem, European Journal of Operational Research.

[11] C. Archetti, M. G. Speranza, D. Vigo, Vehicle Routing: Problems, Methods, and Applications, Vol. 18, SIAM, 2014, Ch. Vehicle routing problems with profits, pp. 273–297.

[12] G. Laporte, S. Martello, The selective travelling salesman problem, Discrete Applied Mathematics 26 (2-3) (1990) 193–207.

[13] B. L. Golden, L. Levy, R. Vohra, The orienteering problem, Naval Research Logistics 34 (3) (1987) 307–318.

[14] C.-W. Chu, A heuristic algorithm for the truckload and less-than-truckload problem, European Journal of Operational Research 165 (3) (2005) 657–667.

[15] M.-C. Bolduc, J. Renaud, F. Boctor, G. Laporte, A perturbation metaheuristic for the vehicle routing problem with private fleet and common carriers, Journal of the Operational Research Society 59 (6) (2008) 776–787.

[16] J.-F. Côté, J.-Y. Potvin, A tabu search heuristic for the vehicle routing problem with private fleet and common carrier, European Journal of Operational Research 198 (2) (2009) 464–469.

[17] J.-Y. Potvin, M.-A. Naud, Tabu search with ejection chains for the vehicle routing problem with private fleet and common carrier, Journal of the Operational Research Society 62 (2) (2011) 326–336.

[18] A. Stenger, D. Vigo, S. Enz, M. Schwind, An adaptive variable neighborhood search algorithm for a vehicle routing problem arising in small package shipping, Transportation Science 47 (1) (2013) 64–80.

[19] S. Irnich, P. Toth, D. Vigo, Vehicle Routing: Problems, Methods, and Applications, Vol. 18, SIAM, 2014, Ch. The Family of Vehicle Routing Problems, pp. 1–33.

[20] S. Martello, P. Toth, Heuristic algorithms for the multiple knapsack problem, Computing 27 (2) (1981) 93–112.

[21] D. Pisinger, S. Ropke, A general heuristic for vehicle routing problems, Computers & Operations research 34 (8) (2007) 2403–2435.

[22] P. Augerat, J. Belenguer, E. Benavent, A. Corberán, D. Naddef, G. Rinaldi, Computational results with a branch and cut code for the capacitated vehicle routing problem, Rapport de recherche- IMAG.

Appendix

A mixed integer linear programming (MILP) formulation

We present a mixed integer linear programming (MILP) formulation for the CCRP. Let the decision variable x_{ij} be 1 if arc (i, j) is used in a flexible carrier route, and 0 otherwise. Furthermore, let the variable y_i be 1 if customer $i \in N$ is outsourced to the fixed carrier, and 0 if it is served by the flexible carrier. The integer variable w represents the number of required transfer trips. The MILP formulation is given below:

$$
\begin{aligned}
\min \quad & \sum_{i\in V}\sum_{j\in V} c_{ij}x_{ij} + Fw \\
\text{s.t} \quad y_i + \sum_{j\in V} x_{ij} &= 1 && \forall i \in N && (1)\\
y_j + \sum_{i\in V} x_{ij} &= 1 && \forall j \in N && (2)\\
\sum_{i\in N} q_i y_i &\leq Q'w && && (3)\\
\sum_{i\in S_r} q_i y_i &\leq E_r && \forall r \in R && (4)\\
\sum_{i\in C, j\notin C} x_{ij} &\geq \left\lceil \frac{\sum_{i\in C} q_i}{Q} \right\rceil && \forall C \subseteq N, C \neq \emptyset && (5)\\
x_{ij} &\in \{0,1\} && \forall i \in V, j \in V && (6)\\
y_i &\in \{0,1\} && \forall i \in N && (7)\\
w &\in \mathbb{Z}_{\geq 0} && && (8)
\end{aligned}
$$

The objective is to minimize the total costs of routing the non-outsourced customers and the cost of transferring the demand of the outsourced customers to the transfer point. Constraints (1) and (2) ensure that a customer is either visited by a single vehicle of the flexible carrier or is outsourced to the fixed carrier. Constraint (3) ensures that the total demand of outsourced customers does not exceed the capacity of the vehicles used for the transfer trips. The selection of customers for outsourcing is constrained by the excess capacity of the fixed carrier routes which is modeled by constraints (4). The constraints (5) ensure that every subtour includes the depot and does not violate the vehicle capacity constraints, and hence represents a feasible route. Constraints (6), (7) and (8) specify the domains of the decision variables.