

# Shared Capacity Routing Problem – An Omni-channel Retail Study

Joydeep Paul<sup>a,\*</sup>, Niels Agatz<sup>a</sup>, Remy Spliet<sup>b</sup>, René de Koster<sup>a</sup>

<sup>a</sup>*Rotterdam School of Management, Erasmus University, Burgemeester Oudlaan 50, 3062PA Rotterdam, The Netherlands*

<sup>b</sup>*Erasmus School of Economics, Erasmus University Rotterdam, Burgemeester Oudlaan 50, 3062PA Rotterdam, The Netherlands*

---

## Abstract

More and more retailers allow customers to order goods online and then pick them up in a store. In this setting, these orders are typically served from a dedicated warehouse. This often means that the stores are visited by different vehicles to replenish the store inventory and to supply the pick-up points. Motivated by a collaboration with an omni-channel grocery retailer in the Netherlands, we study how to best share capacity between the routes associated with these different sales channels. As operational constraints prevent jointly planning the routes, we consider the replenishment routes as fixed when planning the routes to serve the pick-up orders. An order can be transferred to the replenishment route, if capacity allows. We consider the problem of deciding which customer orders to transfer and which to deliver directly such that the total costs are minimized. We present an exact and heuristic approach to solve this problem. Computational experiments on both real-world and artificial instances show that substantial savings can be achieved by sharing vehicle capacity across different channels.

*Keywords:* Vehicle routing; Capacity sharing; Omni-channel retail; Heuristics

## 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

---

\*Corresponding author

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

retailers are expanding their physical presence by opening up regular stores [22]. An omni-channel service model that is increasingly popular is one that allows customers to buy goods online and then pick them up in a store [9]. According to a recent report [13], 64 percent of Europe’s top 500 retailers offer such an in-store pick-up service. A similar trend is seen in the U.S.A [21].

There are different fulfillment strategies for this store pick-up service model. When the number of pick-up 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 [4] [11] use a dedicated warehouse tailored to handling e-fulfillment orders. We focus on this setting in which the pick-up 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 that also serve as pick-up point (PUP) for goods ordered online. The PUPs are supplied from a dedicated e-fulfillment warehouse, while the store inventory is replenished from a traditional warehouse. 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.

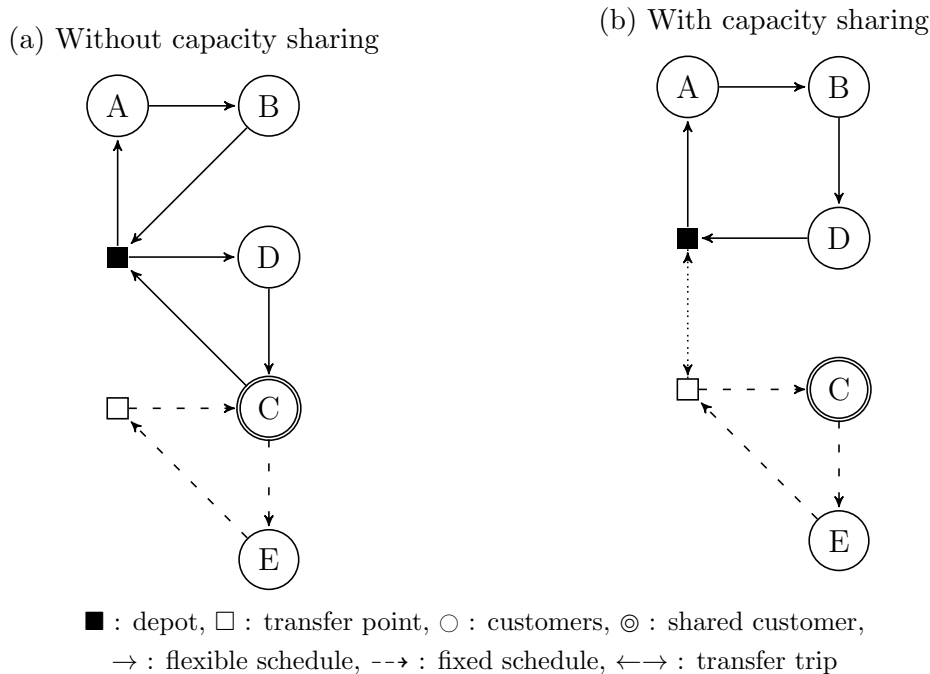
Theoretically, it would be beneficial for the retailer to jointly plan the supply of the pick-up points and the replenishment of the stores. However, this is difficult in practice due to various operational constraints and the cost of synchronizing the different processes. For example, while the replenishment routes need to be planned days in advance to facilitate efficient warehouse operations, the routes to supply the PUPs are planned much later due to their short customer lead-times. Hence, in this paper, we focus on a simple capacity sharing mechanism in which the replenishment routes are fixed in advance and the PUP supply operations can piggyback on those routes.

This works as follows. The retailer fixes the route schedule for the replenishment routes (*fixed* schedule) before planning the routes for the supply of the pick-up points (*flexible* schedule). If there is spare capacity available in the fixed schedule, we can transfer a *shared* customer that is served in both schedules from the flexible schedule to the fixed schedule.

The transfer of the relevant customer demands takes place at the transfer point which is for instance the warehouse associated with the fixed schedule. The retailer incurs additional transfer costs to move the load to this transfer point using vehicles with limited capacity. For the capacity sharing to be beneficial, the transfer costs should be less than the savings in the transport costs.

In Figure 1, we illustrate this capacity sharing opportunity through an example. When there is no capacity sharing, the flexible schedule needs two vehicles to serve its four customers  $A, B, C$ , and  $D$  as shown in Figure 1a. Customer  $C$  is also served in the fixed schedule. The available spare capacity in the fixed schedule makes it possible to *transfer* the shared customer  $C$  from the flexible schedule to the fixed schedule. Figure 1b shows that as a result, only three customers need to be visited in the flexible schedule, reducing both the travel costs and the number of customer visits. To move the demand of the transferred customers from the depot to the transfer point it requires a transfer trip.

Figure 1: Capacity sharing between the fixed and the flexible schedule



In this paper, we introduce the Shared Capacity Routing Problem (SCRCP) which aims to minimize the total costs of the retailer to serve all customers, potentially using the spare

capacity in the fixed schedule. We specifically focus on investigating the benefits of capacity sharing in different settings.

Our contribution is threefold. First, we describe a new capacity sharing strategy motivated by an application in omni-channel retailing. Secondly, we present exact and heuristic approaches to solve the associated planning problem. Finally, we present a numerical study to investigate the benefits of the proposed capacity sharing strategy using both real-world 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 the heuristic approach to solve the problem. Section 7 reports computational results on various instances. Finally, Section 8 summarizes our key findings and provides directions for future research.

## 2. Related literature

Conceptually, the SCRPP is a selective vehicle routing problem 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 [1]. The selective vehicle routing problem is the multi-vehicle version of the selective traveling salesman problem [16] or the orienteering problem [10], 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 SCRPP 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 [5, 3]. Most work on the VRPPC is focussed on the design of heuristics with Tabu search [6, 20] and adaptive variable neighborhood search [23] currently showing the most promising results.

What distinguishes the SCRP from the existing work in the area of selective vehicle routing is that the customers that can be transferred are constrained by the spare capacity in the routes of the fixed schedule. Furthermore, the cost and capacity of transfer trips between the depot and the transfer point also play a critical role in deciding which customers will be transferred.

The concept of sharing vehicle capacity is also relevant when we consider independent carriers in a distribution network. Recent work by Fernández et al. [7] considers the centralized planning in a coalition of carriers in which demands of only a limited number of shared customers can be transferred between the carriers. In the SCRP, we also consider a set of shared customers between two distribution channels of a retailer. However, in our setting, the retailer plans the routes of the fixed schedule in advance and the spare capacity of the fixed schedule can be used to serve some shared customers of the flexible schedule. Unlike our paper, Fernández et al. do not specially consider the transfers between the depots of the different carriers.

Research in the collaboration of carriers mostly focusses on the selection of appropriate collaboration partners and mechanisms for exchanging requests among partners [8]. Co-operative game theory and combinatorial auctions are used for profit sharing in horizontal collaboration among logistic partners [15] [14]. These questions are less relevant in our context as our work is motivated by collaboration within a single retailer. We also primarily focus on the question on how to operationally plan the collaboration in order to attain maximum benefits.

### 3. Problem definition

We model the SCRP on a complete directed graph  $G = (V, A)$ . Here,  $V = \{o\} \cup N$ , where  $o$  is the depot and  $N$  is the set of customer locations. Each customer  $i \in N$  has a demand  $q_i \geq 0$ , which has to be fulfilled from the depot  $o$ . We model two ways in which demand can be fulfilled. Demand of each customer can be fulfilled by direct delivery, and for some customers there is the additional option of fulfilling demand by transferring. Next, we describe these two ways in detail. Note that we do not allow splitting of demand while

servicing a customer, which means that a customer is visited exactly once directly or its demand is fully transferred.

To fulfill demand directly, a sufficient number of vehicles is available, each with capacity  $Q$ . We assume  $Q \geq q_i, \forall i \in N$ . Vehicles are used to drive a *route*, which is a simple cycle in  $G$  starting and ending at the depot, 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 does not exceed the capacity  $Q$ . We refer to the set of routes that we design to fulfill demand directly as the *flexible schedule*. Furthermore,  $c_{ij}$  is the cost of traversing an arc  $(i, j) \in A$ . We assume that  $c_{ij}$  satisfies the triangle inequality.

To fulfill demand by transferring, we are given a *fixed schedule*. A fixed schedule represents a separate routing schedule in which deliveries are made, other than the flexible schedule. Let  $S \subseteq N$  be a set of customers referred to as *shared customers*, they can be thought of as customers that are also visited in the fixed schedule. Only the orders of these shared customers might be transferred to the fixed schedule.

We represent a fixed schedule as the set  $R$  which we refer to as the *fixed routes*, where every fixed route  $r \in R$  corresponds to a (possibly empty) collection of shared customers  $S_r \subseteq S$ . We assume all collections  $S_r$  to be disjoint. Associated with every fixed route  $r \in R$  is a *spare capacity*  $E_r \geq 0$ , representing the leftover capacity in the vehicle associated with fixed route  $r$ . Customers can only be transferred if the spare capacity is not exceeded. That is, a set of customers  $T_r \subseteq S_r$  can only be transferred if the total demand of the customers in  $T_r$  does not exceed the spare capacity  $E_r$ . We refer to such a set  $T_r$  as an *r-transfer*. To represent the transfers to all fixed routes, we define a *transfer-set* as a set of customers  $T \subseteq S$  which is the union of exactly one *r-transfer* per route,  $T = \bigcup_{r \in R} T_r$ .

The demand of the transfer-set customers needs to be transported from the depot to the warehouse associated with the fixed schedule, to transfer the goods to the fixed routes. We refer to this warehouse as the *transfer point*. We use *transfer vehicles* of capacity  $Q'$  to make the transfers. A fixed cost  $F$  is incurred per transfer trip. Based on the practical case that motivated our research, we make the following assumptions with respect to the transfer trips: (i) There is a sufficient number of transfer vehicles available to move the

demands of all transferred customers to the transfer point. (ii) The transfer trips arrive at the transfer point in time to be loaded on to the fixed routes before they depart so we do not have to synchronize the different routes. (iii) Although demand may not be split when serving a customer, it is allowed to split demand of transferred customers on the transfer trips. As a result, for a particular choice of transfer-set  $T$  the total transfer costs are given by  $F \left\lceil \frac{\sum_{i \in T} q_i}{Q'} \right\rceil$ .

As the cost of fixed schedules are exogenous to the model, the total relevant system costs only include the transfer costs and the routing costs of the customers that are not transferred. The objective of the SCRП is to determine a transfer-set and corresponding routes for non-transferred customers so that the total costs are minimized.

As the SCRП reduces to the vehicle routing problem when there is no spare capacity in fixed routes, the SCRП is NP-hard. The appendix provides a mixed integer linear programming (MILP) formulation for the SCRП based on a two-index formulation for the capacitated VRP [12]. In preliminary experiments, we could solve only very small instances with this MILP using the GUROBI solver. In the next section, we discuss theoretical properties of the SCRП which help us to develop our solution strategy for larger instances.

#### 4. Theoretical properties

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

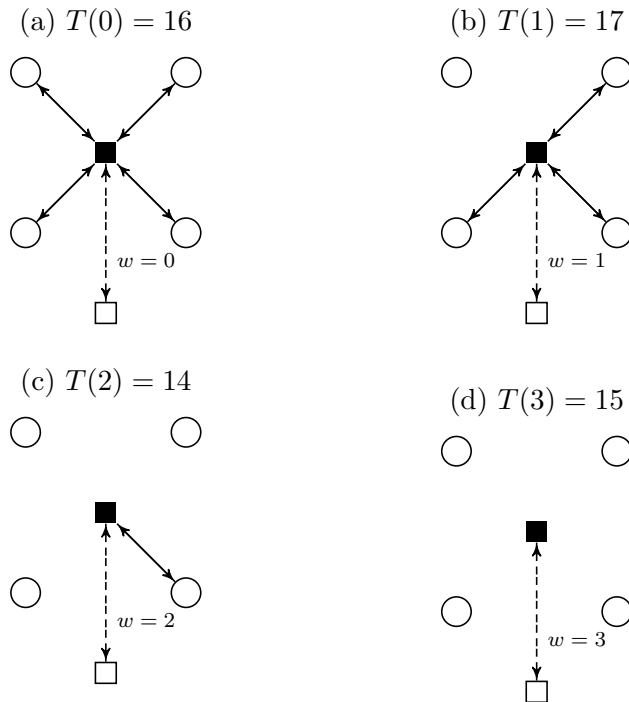
**Proposition 1.** *The optimal cost,  $T(w)$ , of SCRП is in general neither convex nor concave in  $w$ .*

*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 the cost of delivering to each of them from the depot is 4. The

capacities of the vehicles of the flexible schedule and the transfer vehicles are the same, i.e.,  $Q = Q'$ . The transfer cost per trip  $F$  is 5. The spare capacity of the fixed routes is such that all the customers can be transferred. Observe that because demand cannot be split while serving a customer, every non-transferred customer is visited by a separate vehicle.

When  $w = 0$ , no customers are transferred and every customer is visited by a separate vehicle, hence  $T(0) = 16$ . For  $w = 1$ , the optimal decision is to transfer one customer to the fixed schedule, so  $T(1) = 17$ . In case  $w = 2$ , three customers can be transferred, now it follows that  $T(2) = 14$ . Finally, for  $w = 3$ , it is optimal to transfer 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.  $\square$

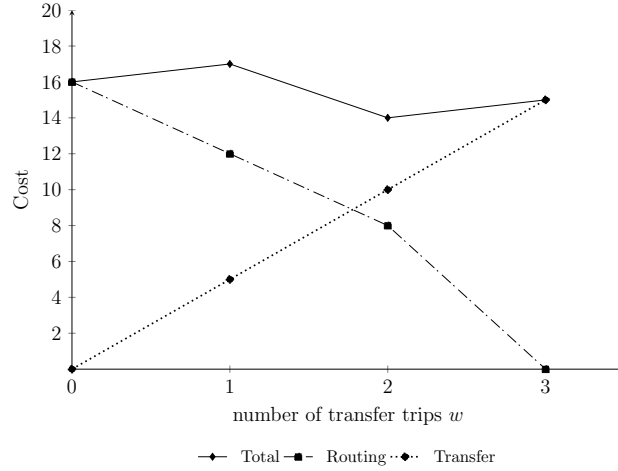
Figure 2: Optimal solutions for our example ( $Q' = Q; q_i = \frac{4}{7}Q; F = 5; c_{oi} = 2, \forall i \in N$ )



■ : depot, □ : transfer point, ○ : customers, ↔: flexible schedule 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 SCRP.*

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

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

*Proof.* The total spare capacity in the fixed routes is given by  $\sum_{r \in R} E_r$ . Since we can split demands in transfer trips, the number of transfer trips required to fully utilize the available spare capacity is  $\left\lceil \frac{\sum_{r \in R} E_r}{Q'} \right\rceil$ . Hence,  $w^* \leq UB_1$ . Similarly, the total transferred demand is limited by  $\sum_{i \in N} q_i$ , yielding  $w^* \leq UB_2$ .

Next, we prove that  $w^* \leq UB_3$ . 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  $w^* \leq \left\lceil \frac{R(0) - R(X)}{F} \right\rceil$ .  $\square$

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

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

**Proposition 3.** *There exists an optimal solution of the SCRП for which the transfer-set is maximal.*

*Proof.* Assuming that the triangle inequality holds, we know that the routing cost is decreasing with the number of transferred customers. Hence, if the transfer-set is not maximal, an additional customer demand can be transferred without increasing the costs.  $\square$

We can now reformulate our problem in the following way:

$$\min_{0 \leq w \leq UB} \left\{ Fw + \min_{T \in T_{max}} R(N \setminus T) \right\}$$

where,  $T_{max}$  is the set of all maximal transfer-sets, 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 transfer-set 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 transfer-sets. Finally, for every max-

imal transfer-set, we solve the vehicle routing problem (VRP) visiting the non-transferred customers. The best found solution is optimal.

For a given maximal transfer-set, the SCRП 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 [12]. We relax the rounded capacity constraints, identify 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 SCRП to terminate the evaluation of certain transfer-sets as follows. If at any stage of the branch-and-cut procedure to solve a VRP, the lower bound plus the transfer costs for the incumbent solution is higher than the current best solution, we discontinue the evaluation of this transfer-set and continue with the next.

## 6. Knapsack-based Heuristic

In the exact approach as described in the previous section, we enumerate all maximal transfer-sets and solve the associated VRP to find the optimal solution. This is not practically feasible for larger instances due to the large number of transfer-sets that need to be evaluated by solving a routing problem. Hence, we develop a heuristic to identify promising transfer-sets and solve the corresponding routing problems. In particular, we present a *knapsack-based* heuristic that aims to find an initial solution by solving a multiple knapsack problem to determine a transfer-set. Subsequently, we implement a local search procedure to improve the initial solution.

### 6.1. Initial solution

Instead of evaluating all maximal transfer-sets, we try to find promising transfer-sets by solving a multiple knapsack problem given the spare capacities in the fixed routes. The main idea is to approximate the savings of transferring customer  $i \in N$  and then find the transfer-set that maximizes the approximate total savings. We use the travel cost from the depot to a customer  $i \in N$ ,  $c_{oi}$  to approximate the savings of transferring a customer  $i$ .

The reason for this is that we expect that it would generally be more advantageous to skip customers that are further away from the depot. Preliminary experiments show that this travel cost based approximation measure provides better results than simply maximizing the number of transferred customers or the associated demand volume.

We formulate the corresponding optimization problem as a multiple knapsack problem [17] where the knapsacks correspond to the capacity constraint on the transfer-set, and the capacity constraints on the  $r$ -transfers. Let the variable  $y_i$  be 1 if customer  $i$  is transferred to the fixed schedule, and 0 otherwise. The problem is formulated as follows:

$$\begin{aligned} \max \quad & \sum_{i \in N} c_{oi} y_i \\ \text{s.t.} \quad & \sum_{i \in S_r} q_i y_i \leq E_r \quad \forall r \in R \end{aligned} \quad (1)$$

$$\sum_{i \in N} q_i y_i \leq Q'w \quad (2)$$

$$y_i \in \{0, 1\} \quad (3)$$

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

As solving a VRP to evaluate the cost of a particular transfer-set is computationally intractable for larger instances, we use our implementation of the adaptive large neighborhood search (ALNS) heuristic by Pisinger and Ropke [19]. This approach uses a local search framework based on simulated annealing and several destroy and repair operators.

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

## 6.2. Improvement phase

To improve the initial solution, we develop a local search heuristic that iterates 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  $r$ -transfer exchange neighborhood that is specific to our problem. The search continues until no more improving  $r$ -transfer exchange is found. For every transfer-set considered during the search, including the initial transfer-set, 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 transferred customers for repositioning. In that case, we do not only consider repositioning customers somewhere in a route but we also consider transferring 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 performed at each iteration.

#### *$r$ -transfer exchange neighborhood*

In an  $r$ -transfer exchange, we exchange an  $r$ -transfer  $T_1$  in the current solution with another  $r$ -transfer  $T_2$  for fixed route  $r \in R$ . We perform this exchange as follows. We remove all customers in  $T_1$  and  $T_2$  from the solution. Next, we transfer the customers in  $T_2$ . All the remaining customers are inserted to the routes of the flexible schedule in random order at the cheapest position. Subsequently, we re-optimize the flexible schedule 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  $r$ -transfer exchange corresponding to the best improvement is performed at each

iteration.

## Diversification

If no more improving moves can be found, we apply 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 specific lower and upper limit which depend on the parameters  $\delta < 1$ ,  $\gamma < 1$ ,  $l$  and  $u$  in the following way:

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

This *destroy* operation is similar to the *destroy* operation in the ALNS heuristic by Pisinger and Ropke [19].

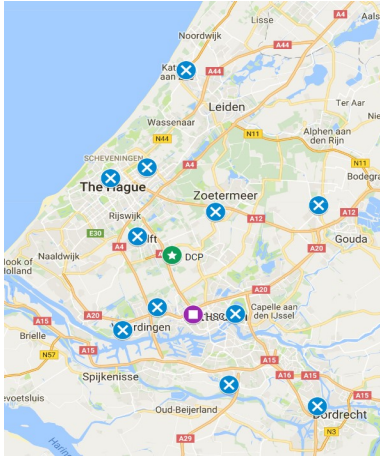
During the repair stage, a customer can either be transferred to the fixed schedule or served in the flexible schedule. If it is feasible to be transferred, we assign it to the fixed schedule with a probability  $\rho$ , otherwise we insert the customer at the first position of the first route of the flexible schedule with sufficient capacity.

## 7. Computational study

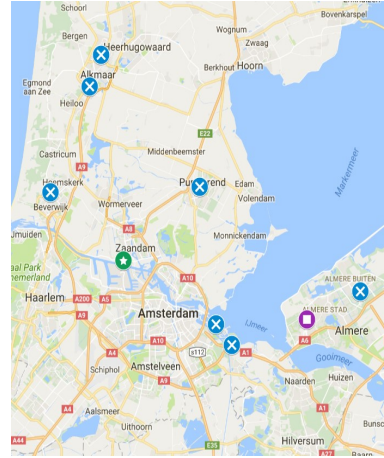
In this section, we report the results of our computational experiments. The goal of these experiments is to assess the quality of our heuristics and the benefits of sharing the capacity of the fixed schedule with the flexible schedule under different settings. All algorithms are coded in JAVA and Gurobi 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 capacity sharing strategy, we apply our model to the distribution network of a large omni-channel grocery retailer in the Netherlands. Some of the retailer’s grocery stores also serve as PUP 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 Southwest region (SW) where there are eleven stores with a PUP that are served in both the channels. Similarly, Figure 4b shows the seven PUP stores that are served in both channels in the Northwest region (NW). In this case, the store replenishment takes place before the in-store pick-up points open to the customers, so all transfers are time feasible.

The average store replenishment demand is approximately 40 roll cages. The average demand of a PUP store is around 6 roll cages for the SW case and 10 roll cages for the NW case. All trucks have a capacity of 50 roll cages. The average spare capacity of the replenishment routes is approximately 20%. We set the transfer cost per trip equal 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 capacity sharing, we determine the optimal routing costs by solving a VRP in which all *shared* customers, i.e., the stores that have a PUP, are visited in the flexible schedule.

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 capacity sharing for the SW and NW cases. We report the savings in routing cost as compared to the costs without capacity sharing while the savings in the number of customer visits is relative to the number of customers (i.e., stores) in the instances.

Table 1: Impact of capacity sharing in real-world instances  
( $F = 20.8$  for SW,  $F = 76$  for NW)

Instance	Days with capacity sharing	$\Delta$ Transport cost* (%)	$\Delta$ Customer-visit* (%)
SW - 11 stores	10/10	33.4	60.9
NW - 7 stores	4/10	1.6	60.7

\*average reduction for instances with capacity sharing

Table 1 shows that there are 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 demand by sharing capacity 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 the routing costs savings in the NW case are offset by the higher transfer costs. Another reason is that the higher demand per store in the NW case limits the number of possible transfers. In both cases, we do observe around 60% reduction in the number of customer visits for the days with capacity sharing.

Sharing capacity also helps to increase the fill rate of the vehicles that perform the store replenishments. In particular, the fill rate of the vehicles of the fixed schedule increases by 7.4% and 12.1% in the SW and NW instances respectively.

While the current instances are small enough to be solved with 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 and test our heuristic.

## 7.2. Generation of artificial instances

We generate artificial instances based on the capacitated VRP instances of [18]. In particular, we use these instances to represent the set of shared customers that is served



in both the flexible and the fixed schedule. For the flexible schedule, we use the customer demand and vehicle capacities as given in these instances and assume the same capacities for the vehicles for the transfer trips. The best known solutions for these instances represent the benchmark solutions for the situation without capacity sharing.

To create the fixed routes, we specify a maximum number of customers per route (*RouteCap*). We then solve a VRP with the ALNS heuristic as described in Section 6.1 to obtain the fixed routes. To generate the spare capacity for each fixed route, we specify the maximum number of customers that can be transferred to each fixed route (*MaxTransfer*). We do this by setting the spare capacity of a route equal to the sum of the demands of the *MaxTransfer* smallest customers from the flexible schedule corresponding to that route. We set the costs per transfer trip to half of the maximum distance between two locations in the graph.

### 7.3. Performance of the Knapsack-based heuristic

To evaluate the performance of our heuristic, we use the VRP instance [2, 18] of size 32 (including the depot) for which we are able to find optimal solutions using the exact approach described in Section 5. We test our heuristic on different instances generated using different parameter values. In particular, we consider  $RouteCap = 2, 3, \dots, 8$  and relative to this capacity, we use  $MaxTransfer = 1, 2, \dots, \lfloor \frac{RouteCap}{2} \rfloor$ . This means that the maximum number of customers that can be transferred to a fixed route is less than half of the customers in that route. We use three performance measures to evaluate the heuristic: average optimality gap, maximum optimality gap and number of times the optimal solution is found.

Table 2 provides the parameter values that are used in the knapsack-based heuristic. The values for the parameters related to the ALNS heuristic are used as reported in [19].

Table 2: Parameter settings of the knapsack-based heuristic

Parameters	Description	Values
$\delta$	lower bound parameters on the number	0.1
$l$	of customers to be removed in the destroy phase	30
$\gamma$	upper bound parameters on the number	0.4
$u$	of customers to be removed in the destroy phase	60
$\rho$	probability of transferring a customer in the repair stage	0.5
$I$	number of iterations of the improvement phase	100

Table 3: Performance of the knapsack-based heuristic ( $|V| = 32, Q = Q' = 100$ )

<i>RouteCap</i>	<i>MaxTransfer</i>	Total possible transfers	Optimality gap (%)	Solution time (s)
2	1	16	1.5	15.1
3	1	11	0.0	19.3
4	1	8	0.0	25.2
	2	16	0.0	28.6
5	1	7	0.0	24.9
	2	13	0.0	18.5
6	1	6	0.0	27.0
	2	11	0.0	19.7
	3	16	0.1	18.9
7	1	5	0.0	34.0
	2	10	0.0	21.1
	3	15	0.0	19.8
8	1	4	0.0	28.7
	2	8	0.0	36.2
	3	12	0.8	31.1
	4	16	0.1	31.1
		Average	<b>0.2</b>	<b>25.0</b>
		Maximum	<b>1.5</b>	<b>36.2</b>
		No. of times optimum found	<b>12/16</b>	

In Table 3, we report the statistics of the solutions obtained by the heuristic for the 16 instances. For each instance, we also report the total maximum number of customers that can be transferred to the fixed schedule.

The table shows that our heuristic provides good quality solutions in reasonable time. The average optimality gap is 0.2%, with a maximum gap of 1.5%. The heuristic finds the

optimal solution in 12 out of the 16 instances.

#### 7.4. Savings by capacity sharing across different instances

Next, we present the results of the experiments with larger instances using the Knapsack-based heuristic. Similar to Section 7.3, we generate the fixed routes for these instances using  $RouteCap = 2, 3, 4, 5, 6, 7, 8$  and  $MaxTransfer = 1, 2, \dots, \lfloor \frac{RouteCap}{2} \rfloor$ . The total costs of the flexible schedule without capacity sharing and the transfer costs per trip for the instances are given in Table 4.

Table 4: Description of the instances

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 5 shows the relative savings in transport costs and the number of customer-visits due to capacity sharing as compared to the setting without capacity sharing across all instances under different settings. We observe that for a given  $RouteCap$ , the savings increase with the spare capacity ( $MaxTransfer$ ). Interestingly, the transport costs savings are not proportional to the savings in customer-visits. This is because transferring a single customer that is further away from the depot leads to more transport cost savings than two nearby customers. The instance in which no savings can be achieved indicate that capacity sharing is not always beneficial. That is, even if transfers are feasible, the transfer costs may outweigh the savings in the routing costs.

We see average computation times up to 300 seconds for the largest instance A-n80-k10. Although the number of feasible solutions increases with the spare capacity, we do not see any clear trend with respect to the solution times in this aspect. Note that the improvement phase drives the longer solution times as finding an initial solution is fast in all settings.

Table 5: Savings (%) by capacity sharing for larger instances

Route	Max Transfer	A-n32-k5			A-n48-k7			A-n64-k9			A-n80-k10		
		$\Delta$ Transport cost (%)	$\Delta$ Customer -visit (%)	Solution time (s)	$\Delta$ Transport cost (%)	$\Delta$ Customer -visit (%)	Solution time (s)	$\Delta$ Transport cost (%)	$\Delta$ Customer -visit (%)	Solution time (s)	$\Delta$ Transport cost (%)	$\Delta$ Customer -visit (%)	Solution time(s)
2	1	9.9	41.9	15.1	18.8	44.7	31.3	17.7	50.8	82.7	17.8	46.8	255.6
	3	1	3.6	35.5	19.3	11.5	27.7	133.4	8.9	33.3	97.3	10.2	25.3
4	1	1.6	25.8	25.2	5.3	25.5	92.5	6.6	23.8	147.7	6.6	25.3	299.7
	2	11.3	41.9	28.6	20.0	44.7	42.1	16.2	42.9	69.6	17.1	48.1	196.9
5	1	0.0	0.0	24.9	2.6	21.3	93.0	5.1	20.6	203.0	5.5	20.3	285.2
	2	5.9	38.7	18.5	13.8	29.8	88.0	13.0	39.7	111.5	11.3	35.4	187.2
6	1	0.4	19.4	27.0	0.0	0.0	109.8	3.7	17.5	160.0	4.4	17.7	404.4
	2	3.9	35.5	19.7	11.6	29.8	51.5	11.2	27.0	259.8	9.0	29.1	197.7
	3	11.6	32.3	18.9	20.9	38.3	84.7	18.9	41.3	226.1	16.5	44.3	131.9
7	1	0.0	0.0	34.0	0.0	0.0	104.5	1.7	14.3	608.5	3.3	15.2	495.0
	2	2.7	32.3	21.1	9.7	25.5	86.6	7.6	23.8	170.0	8.8	27.8	223.9
	3	11.1	41.9	19.8	15.2	31.9	87.8	15.5	33.3	231.8	12.9	41.8	321.1
8	1	0.0	0.0	28.7	0.0	0.0	84.1	0.4	12.7	170.8	1.4	12.7	357.7
	2	1.5	19.4	36.2	4.8	19.1	76.9	5.8	25.4	208.0	7.2	24.1	268.9
	3	6.7	32.3	31.1	12.2	23.4	106.8	10.5	28.6	250.0	10.2	35.4	340.3
	4	12.4	38.7	31.1	22.4	36.2	75.1	19.6	34.9	206.3	17.2	36.7	496.0
Average		5.2	27.2	25.0	10.5	24.9	84.2	10.1	29.4	200.2	10.0	30.4	298.9

$\Delta$  denotes reduction here

### 7.5. Impact of service costs

Thus far, we have primarily focussed on the costs savings related to reducing the total system-wide travel costs. However, there may be other benefits associated with visiting a customer by one vehicle instead of two vehicles. We will refer to these non-distance related costs as service costs. By visiting the customer once instead of twice, it may be possible to save time, e.g., time associated with finding a parking space or waiting at the customer. In this section, we investigate the impact of the potential service cost savings on the solutions of the SCRP.

In our analysis, we focus solely on the service costs that can be avoided by transferring a customer to the fixed schedule. Let  $\tau$  represent this ‘avoidable’ service costs per customer. For example, if the total service costs of serving a customer by two vehicles separately is 20 and costs of serving all demands of a customer from both schedules by one vehicle is 15, we can save  $\tau = 5$ . To normalize the results, we set the value of  $\tau$  to a percentage of the transfer cost per trip. We use our heuristic to run the experiments on instance A-n48-k7 using  $RouteCap=6$  and  $MaxTransfer=3$ .

Table 6: Impact of service costs on savings ( $Q = Q' = 100, F = 60$ )

$\tau$ (%)	# of Transferred customers	$\Delta$ Transport cost (%)	$\Delta$ Service cost (%)
0	18	20.9	0.0
10	21	15.6	9.3
20	21	12.9	15.4
30	22	10.2	20.6
40	23	8.0	25.0
50	23	7.4	27.8

$\Delta$  denotes reduction here

In Table 6, we show the solution for different values of  $\tau$ . To allow a fair comparison with the earlier results, we report the transport cost savings and service cost savings separately. As expected, we see that the number of transfers increase with the potential service cost savings  $\tau$ . The results clearly show the trade-off between the transport costs and the service costs. That is, when  $\tau$  increases, it becomes more beneficial to transfer customers even if

this means increasing the routing costs.

Overall, the benefits of capacity sharing increase with the potential service cost savings.

## 8. Conclusion

This paper studies capacity sharing in an omni-channel retail setting. Motivated by a practical problem, we introduce the shared capacity routing problem (SCRCP). The presented capacity sharing strategy enables the retailer to make use of the spare capacity in its transport operations to reduce the transport costs and the number of customer visits. We show that our heuristic provides good quality solutions in a reasonable amount of time.

The computational study on the real-life case suggest potential transport cost savings between 2% and 33% by better using the available vehicle capacities in the system. The results show that the transfer costs and the spare capacity are the main drivers of the potential benefits of capacity sharing. The benefits increase with the spare capacity and decrease when the transfer costs increase. Potential service cost savings may further increase the benefits of capacity sharing.

As we are the first to work on this problem, there are still many directions for future research in this area. One potential future research direction is to develop exact solution procedures that can solve larger problem instances. Moreover, an interesting extension of the problem is to consider multiple transfer points instead of a single one. The capacity sharing strategy can also be extended to settings that involve multiple companies, which gives rise to questions related to profit sharing.

## References

- [1] Archetti, C., Speranza, M. G., Vigo, D., 2014. Vehicle Routing: Problems, Methods, and Applications. Vol. 18. SIAM, Ch. Vehicle routing problems with profits, pp. 273–297.
- [2] Augerat, P., 1995. Approche polyédrale du problème de tournées de véhicules. Ph.D. thesis, Institut National Polytechnique de Grenoble-INPG.
- [3] Bolduc, M.-C., Renaud, J., Boctor, F., Laporte, G., 2008. A perturbation metaheuristic for the vehicle routing problem with private fleet and common carriers. *Journal of the Operational Research Society* 59 (6), 776–787.

- [4] Bose, N., 2016. 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>.
- [5] Chu, C.-W., 2005. A heuristic algorithm for the truckload and less-than-truckload problem. *European Journal of Operational Research* 165 (3), 657–667.
- [6] Côté, J.-F., Potvin, J.-Y., 2009. A tabu search heuristic for the vehicle routing problem with private fleet and common carrier. *European Journal of Operational Research* 198 (2), 464–469.
- [7] Fernández, E., Roca-Riu, M., Speranza, M. G., 2017. The shared customer collaboration vehicle routing problem. *European Journal of Operational Research*.
- [8] Gansterer, M., Hartl, R. F., 2017. Collaborative vehicle routing: a survey. *arXiv preprint arXiv:1706.05254*.
- [9] Gao, F., Su, X., 2016. Omnichannel retail operations with buy-online-and-pick-up-in-store. *Management Science* 63 (8), 2478–2492.
- [10] Golden, B. L., Levy, L., Vohra, R., 1987. The orienteering problem. *Naval Research Logistics* 34 (3), 307–318.
- [11] Hübner, A., Kuhn, H., Wollenburg, J., 2016. Last mile fulfilment and distribution in omni-channel grocery retailing: a strategic planning framework. *International Journal of Retail & Distribution Management* 44 (3), 228–247.
- [12] Irnich, S., Toth, P., Vigo, D., 2014. *Vehicle Routing: Problems, Methods, and Applications*. Vol. 18. SIAM, Ch. The Family of Vehicle Routing Problems, pp. 1–33.
- [13] Jindal, I., 2017. A performance ranking of Europe's Top500 ecommerce and multichannel retailers. <http://viewer.zmags.com/publication/5f09e229#/5f09e229/22>.
- [14] Krajewska, M. A., Kopfer, H., 2006. Collaborating freight forwarding enterprises. *OR spectrum* 28 (3), 301–317.
- [15] Krajewska, M. A., Kopfer, H., Laporte, G., Ropke, S., Zaccour, G., 2008. Horizontal cooperation among freight carriers: request allocation and profit sharing. *Journal of the Operational Research Society* 59 (11), 1483–1491.
- [16] Laporte, G., Martello, S., 1990. The selective travelling salesman problem. *Discrete Applied Mathematics* 26 (2-3), 193–207.
- [17] Martello, S., Toth, P., 1981. Heuristic algorithms for the multiple knapsack problem. *Computing* 27 (2), 93–112.
- [18] NEO - Networking and Emerging Optimization, 2013. Augerat et al. <http://neo.lcc.uma.es/vrp/vrp-instances/capacitated-vrp-instances/>.
- [19] Pisinger, D., Ropke, S., 2007. A general heuristic for vehicle routing problems. *Computers & Operations*

research 34 (8), 2403–2435.

- [20] Potvin, J.-Y., Naud, M.-A., 2011. Tabu search with ejection chains for the vehicle routing problem with private fleet and common carrier. *Journal of the Operational Research Society* 62 (2), 326–336.
- [21] Rosenblum, P., Kilcourse, B., 2013. Omni-channel 2013: The long road to adoption, RSR 2013 benchmark report, Retail Systems Research.
- [22] Speculations, G., Feb 2016. Why Would Amazon Open Physical Stores?  
URL <https://www.forbes.com/sites/greatspeculations/2016/02/11/why-would-amazon-open-physical-stores/#44c11c2f964d>
- [23] Stenger, A., Vigo, D., Enz, S., Schwind, M., 2013. An adaptive variable neighborhood search algorithm for a vehicle routing problem arising in small package shipping. *Transportation Science* 47 (1), 64–80.

## Appendix

### A mixed integer linear programming formulation

We present a mixed integer linear programming (MILP) formulation for the SCRP. Let the decision variable  $x_{ij}$  be 1 if arc  $(i, j)$  is used in a route of the flexible schedule, and 0 otherwise. Furthermore, let the decision variable  $y_i$  be 1 if customer  $i \in N$  is transferred to the fixed schedule, and 0 if it is served in the flexible schedule. The integer variable  $w$  represents the number of required transfer trips. Let  $u_i$  indicate the accumulated demand already distributed by the vehicle when arriving at customer  $i \in N$ . The MILP formulation is given below:



$$\begin{aligned}
\min \quad & \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} + Fw \\
\text{s.t.} \quad & y_i + \sum_{j \in V} x_{ij} = 1 & \forall i \in N & (1) \\
& y_j + \sum_{i \in V} x_{ij} = 1 & \forall j \in N & (2) \\
& \sum_{i \in N} q_i y_i \leq Q'w & & (3) \\
& \sum_{i \in S_r} q_i y_i \leq E_r & \forall r \in R & (4) \\
& u_i - u_j + Qx_{ij} \leq Q - q_j & \forall i \in N, j \in N & (5) \\
& x_{ij} \in \{0, 1\} & \forall i \in V, j \in V & (6) \\
& y_i \in \{0, 1\} & \forall i \in N & (7) \\
& w \in \mathbb{Z}_{\geq 0} & & (8) \\
& q_i \leq u_i \leq Q & \forall i \in N & (9)
\end{aligned}$$

The objective is to minimize the total cost of routing the non-transferred customers and the cost of transferring the demand of the transferred customers to the transfer point. Constraints (1) and (2) ensure that a customer is either visited by a single vehicle of the flexible schedule or is transferred to the fixed schedule. Constraint (3) ensures that the total demand of transferred customers does not exceed the capacity of the vehicles used for the transfer trips. The selection of customers for transferring is constrained by the spare capacity of the fixed 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), (8) and (9) specify the domains of the decision variables.