

Contents lists available at [SciVerse ScienceDirect](#)

European Journal of Operational Research

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

A new VRPPD model and a hybrid heuristic solution approach for e-tailing

Seda Yanik^a, Burcin Bozkaya^{b,*}, Ronan deKervenoael^b

^a Istanbul Technical University, Industrial Engineering Department, Turkey

^b Sabanci University, Sabanci School of Management, Turkey

ARTICLE INFO

Article history:
Available online xxx

Keywords:
VRP with pickup and deliveries and time windows
Hybrid heuristic
Genetic Algorithm
GIS

ABSTRACT

We analyze a business model for e-supermarkets to enable multi-product sourcing capacity through co-opetition (collaborative competition). The logistics aspect of our approach is to design and execute a network system where “premium” goods are acquired from vendors at multiple locations in the supply network and delivered to customers. Our specific goals are to: (i) investigate the role of premium product offerings in creating critical mass and profit; (ii) develop a model for the multiple-pickup single-delivery vehicle routing problem in the presence of multiple vendors; and (iii) propose a hybrid solution approach. To solve the problem introduced in this paper, we develop a hybrid metaheuristic approach that uses a Genetic Algorithm for vendor selection and allocation, and a modified savings algorithm for the capacitated VRP with multiple pickup, single delivery and time windows (CVRPMPDTW). The proposed Genetic Algorithm guides the search for optimal vendor pickup location decisions, and for each generated solution in the genetic population, a corresponding CVRPMPDTW is solved using the savings algorithm. We validate our solution approach against published VRPTW solutions and also test our algorithm with Solomon instances modified for CVRPMPDTW.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

In a market-driven e-grocery setting, fulfilling consumer demand for diverse and premium products is a challenge. A key strategy to increase the number of customers served as well as the margin and the basket size in many businesses is to offer premium services and a variety of products that meet the needs and desires of consumers. Yet, most e-groceries have not been able to elevate a business from providing low value ‘staple’ and bulky goods to fulfilling the diverse consumer demand for premium (e.g. organic) or premium-priced (e.g. wine, special gifts) products for special occasions (e.g. birthday, anniversary). In this paper, we analyze a business model for e-supermarkets to enable multi-product sourcing capacity through co-opetition (collaborative competition). The logistics aspect of our approach is to design and execute a network system where (premium) goods are acquired from possibly different vendors at multiple locations in the supply network and delivered in a single visit to each customer. Our specific goals are to: (i) investigate the role and business value of premium product offerings and advanced logistics in creating critical mass and profit; (ii) develop a model for a variant of the vehicle routing problem (VRP) with pickup and delivery and time windows; and (iii) propose a hybrid metaheuristic solution approach.

The proposed business model addresses a co-opetitive operation between an e-supermarket chain and premium product vendors. In this setting, the product range offered by the e-supermarket includes the SKUs (stock keeping units) in the brick-and-mortar grocery stores and the premium products at a set of external third-party provider/vendor locations. We distinguish between the former and latter category of products using the terms *standard* and *premium*, where the first group is provided only by the brick-and-mortar store; the second group, only by external vendors. We assume that neither standard nor premium goods require special handling for cases such as perishability or special storage such as refrigerating, special rack or pallet. Thus, we include no special constraints regarding the transportation or storage of goods in the vehicle. Furthermore, we set a same-day order cycle equivalent to a couple of hours, based on the practical operation cycles within predefined time slots.

The routing problem of interest consists of two components: (a) selection of vendors and their allocation to customers who have ordered premium products in their e-basket; (b) routing of the vehicles that serve the customers by picking up customers’ premium products from external vendors and consolidating them with selected in-store products for delivery as a single complete basket. For each customer, a multiple pickup and single delivery combination is defined by assigning specific vendors to that customer. We refer to this problem, which is a variant of the classical capacitated vehicle routing problem with pickup and deliveries as well as time windows, as CVRP with multiple pickup, single delivery and time

* Corresponding author. Tel.: +90 216 483 9687.

E-mail addresses: sedayanik@itu.edu.tr (S. Yanik), bbozkaya@sabanciuniv.edu (B. Bozkaya), dekervenoael@sabanciuniv.edu (R. deKervenoael).

windows (CVRPMPDTW). We formulate this problem as a mixed integer programming model with an objective of minimizing total transportation costs. In our model, a given fleet of vehicles originates at the store location from which standard in-store e-delivery products are assembled and shipped. The vehicles visit assigned vendor locations to pick up premium products before delivering the complete basket to a customer. Additional constraints make sure vehicle capacities are not violated and the customer delivery time windows are honored. This type of problem and the corresponding model, to the best of our knowledge, has not been previously studied in the literature.

To solve the CVRPMPDTW, we have developed a hybrid meta-heuristic approach using a Genetic Algorithm for vendor selection and customer–vendor allocation, and a modified savings algorithm for the CVRPMPDTW. The proposed Genetic Algorithm guides the search for optimal vendor pickup location decisions; then for each generated solution in the genetic population, the savings algorithm solves a corresponding CVRPMPDTW instance. We first validate our solution methodology against published solutions of Solomon VRPTW instances. Then we test our methodology on a random dataset based on modified versions of these Solomon instances. Next, we test the economic viability of our business model using a real dataset in the metropolis of Istanbul, Turkey where the use of e-grocery has had a marked acceleration both in scope and scale (Yanik et al., 2010; deKerpenoael et al., 2011). We introduce the concept of profitability by assuming basket sizes and profit margins for standard as well as premium products. We then perform a sensitivity analysis based on the sales parameters and transportation costs. Results show that the increased cost of logistics operations due to inclusion of premium vendors *can* be potentially offset and translated into higher gross margins, with increased product profit margins and basket sizes, all of which demonstrate a valid business case.

The rest of the paper is as follows: in Section 2, we present an overview of the related literature and point out the fact that our problem is a new variant of the VRP with pick up and deliveries and time windows. In Section 3, we present our model; in Section 4, our solution methodology. Section 5 provides the results of our numerical analysis and case study. Finally, in Section 6, we conclude with further remarks on applicability of our proposed business model.

2. Literature

The problem we study in this paper is considered as a variant of capacitated VRP with pickup and deliveries and time windows (CVRPPDTW) in the literature. A classification of VRP with pickup and delivery has been presented in the review paper of Parragh et al. (2008a,b). They first define a main class of problems, which deal with the transportation of goods from the depot to linehaul customers and from backhaul customers to the depot. Four subtypes of problems have been identified in this class. In the VRP with clustered backhauls, all linehauls are completed before backhauls. VRP with mixed linehauls and backhauls allows any sequence. In the VRP with divisible (simultaneous) delivery and pickup, customers can be visited twice (simultaneously) both for delivery and pickup. The second main class is identified as classical to all those types of problems where goods are transported between pickup and delivery locations. This class comprises problems in which pickup and delivery points may be paired or unpaired. A special type of the paired pickup and delivery VRP that deals with passenger transportation is identified as Dial-a-Ride problem in the literature.

The CVRPMPDTW arising from the business case defined in our Introduction is basically a paired pickup and delivery problem but differs from the classical problems defined in the literature. In our

problem setting, we consider multiple types of commodities. The vendor pickup locations are specialized based on commodities; that is, each pickup location can only supply one type of commodity. Yet, there are multiple vendor locations in the network, offering each type of commodity, and choosing one or more vendor locations to supply a particular type of commodity to a specific customer is an assignment-type problem. Furthermore, the transportation of goods from the originating store to customer delivery points exists, in addition to the transportation of the paired pickup and deliveries. In other words, the vehicles leave the store with a load of standard goods for which the target delivery points are customer addresses.

The complexity of VRPPD problems is higher than the classical VRP, which is an NP-hard problem. The single vehicle case, namely the traveling salesman problems (TSPs), has been studied widely. Yet it provides insights into the complex problem structure and allows developing solution concepts. These approaches have then been used to develop exact and approximate algorithms for VRPs (Toth and Vigo, 2002). Such an instrument is used similarly for VRPPDs in the literature in order to investigate the single vehicle case first, and then use the acquired knowledge for the search of the multi-vehicle problem solution.

The pickup and delivery traveling salesman problem (PDTSP), which is a generalization of the well-known TSP where each customer provides or requires a given non-zero amount of product served by a single vehicle, has also been investigated in the literature along with the exact and heuristic solution methodologies developed (Hernandez-Perez and Salazar-Gonzalez, 2004). The methodologies offering exact solution algorithms are branch-and-bound (Kalantari et al., 1985), branch-and-cut (Balas et al., 1995), and dynamic programming (Desrosiers et al., 1986; Bianco et al., 1994). PDTSPs are defined in a setting where the goods to be delivered are of a different type with respect to the goods to be collected. A similar problem to PDTSPs is the one-commodity Pickup and Delivery Traveling Salesman Problem (1-PDTSP) where the set of customer locations is partitioned into pickup and delivery customers to be served by a single vehicle based at the depot for a single commodity (Berbeglia et al., 2007). In the literature, the exact solutions for 1-PDTSPs have been also tackled with branch-and-cut procedures (Hernandez-Perez and Salazar-Gonzalez, 2004, 2007). Besides, an exact algorithm for the TSP with clustered backhauls (the case where all deliveries are completed before pickups) has been presented by Baldacci et al. (2003) using an additive lower bounding procedure which can be embedded within a branch-and-bound framework similar to the procedure of Fischetti and Toth (1989). On the other hand, heuristics serve as an alternative for solving the PDTSP. Different algorithms such as genetic algorithms, tabu search, local search and insertion methods are developed in the literature (Pankratz, 2005; Landrieu et al., 2001; Van Der Bruggen et al., 1993; Lu and Dessouky, 2006).

Efforts on developing efficient solution algorithms for VRPPD have grown tremendously in the last years. Many literature review papers examine the exact and heuristics solution techniques of VRPPD (Desaulniers et al., 2002; Cordeau and Laporte, 2003a; Nagy and Salhi, 2005; Gendreau et al., 2008; Cordeau and Laporte, 2007; Berbeglia et al., 2007; Cordeau et al., 2008). The most commonly used heuristics to deal with the VRPPD include insertion procedures (Jaw et al., 1986), cluster-first, route-second methods (Cullen et al., 1981; Bodin and Sexton, 1986; Dumas et al., 1989; Desrosiers et al., 1991; Toth and Vigo, 1996; Borndörfer et al., 1997), tabu search (Toth and Vigo, 1997; Nanry and Barnes, 2000; Caricato et al., 2003; Cordeau and Laporte, 2003b; Attanasio et al., 2004; Montane and Galvao, 2006; Melachrinoudis et al., 2007), genetic algorithms (Rekiek et al., 2006; Ganesh and Narendran, 2007), simulated annealing (Hart, 1996; Li and Lim, 2001) and ant colony optimization (Doerner et al., 2001, 2003). In

addition, Bent and van Hentenryck (2006) have developed a hybrid heuristic for the VRPPDTW. In their two-stage heuristic, the first stage applies simulated annealing (SA) to minimize the number of routes, while the second stage minimizes the total route length through large neighborhood search.

Another approach in the VRPPD is to split the vehicles into pickup vehicle fleet and delivery vehicle fleet. In order to maintain an interaction between these two fleets, transfers are introduced during the delivery phase. Lin (2008) has compared a classical VRPPD model to a model where transfers take place between the vehicles at the depot. In another study, Thangiah et al. (2007) have introduced transfer points in the network different from the depot. They allow transfers between the vehicles and assume a divisible delivery to the customer, which leads to multiple numbers of vehicles delivering goods to the customer.

The problem defined in our paper arises from a practical e-delivery setting. In this setting, transfers are not allowed between vehicles and each customer must be visited only once by a single vehicle. However, customer orders may be picked up from the sourcing store as well as the external suppliers (vendor pickup locations) within the network. In this case, the vehicles start their routes with a load of goods available at the sourcing store. The vendor pickup points are matched to customers ordering premium goods, and the vehicles stop at the matching vendor pickup locations to pick up the premium goods ordered. An assignment of a pickup location to each customer is maintained in the solution methodology as well as the route sequence; while the minimum cost solution is sought in the presence of pickup and delivery location precedence, time window and capacity constraints. Based on our literature review, we find that this variant of the CVRPPDTW has not been studied before. However, as the popularity of e-shopping has increased and led to a growth in e-delivery models, problems related to e-delivery have recently attracted attention in the literature. One such study on e-delivery location and routing problem is by Aksen and Altinkemer (2008). The authors formulate a model for store selection for delivery and routes based on the selected stores for online orders. Provided that their study includes a location and routing aspect of the e-delivery setting, the routing part does not contain any pickup operations, which means they are dealing with a classical VRP problem. Besides, the location aspect in their study is the search for the start point of the route whereas in our case we look for the best assignment between the pickup and delivery points so that the total distribution cost is minimized.

In this study, we also aim to show that the case we have described is a viable business model for e-supermarkets in order to create further value added through extra services and a more efficient use of the geographical context in large metropolises. The mathematical model we have developed is a new variant of the CVRP with multiple pickup and deliveries, and time windows. Though we are able to formulate this problem as a mixed integer linear programming model, we resort to a hybrid metaheuristic approach due to the complex nature of the problem.

3. Model

In our proposed model, customers are served with standard products shipped from a sourcing store, whereas the premium products are picked up from the external vendor location(s). Customer delivery is accomplished only after all the pickups of the premium products are made and consolidated with the standard products in a vehicle designated for service. The single delivery to the customer is scheduled during the time window of customer's choice.

We use the following notation for our model formulation presented below. V is the set of vendor (pickup) locations, I is the set of customer (delivery) locations, $D = \{d\}$ is the depot (in case of supermarket e-delivery, depot is the sourcing store). The network

is defined with these locations where, $N = V \cup I$, $A = N \cup D$. The set of customers having premium product orders is denoted by $I_p \subseteq I$. P denotes the set of premium products served to the customers. The premium products and their suppliers are defined by a compatibility matrix denoted by e_{vp} , whereas the quantity to be picked-up from vendor v for customer i is denoted by q_{iv} and the quantity to be loaded onto a vehicle for customer i from the depot is denoted by q_{id} . O_{ip} is a binary indicator taking the value 1 if customer i has an order for premium product p and 0 otherwise.

K is the set of vehicles and C_k denotes the vehicle capacity. The start and the end time of customer i 's time window is denoted by a_i and b_i . The time needed to serve customer i is s_i , and the time needed for a pickup at vendor v is s_v . The time needed for vehicle k to travel from location i to location j is denoted by t_{ijk} , and the associated cost is c_{ijk} . The fixed cost of using vehicle k is f_k . Finally, M represents a sufficiently large value used to ensure only one of the two constraints on pickup-delivery synchronization is in effect.

The decision variables of the model to optimize the routes and the sequence of the stops for the pickup and delivery are as follows:

- x_{ijk} : 1 if vehicle k traverses from location i to location j , otherwise 0; $i, j \in A$, $k \in K$
- z_{ivk} : 1 if customer i is served from vendor v by vehicle k , otherwise 0; $i \in I_p$, $v \in V$, $k \in K$
- y_k : 1 if vehicle k is used, otherwise 0; $k \in K$
- T_{ik} : time for vehicle k to arrive at location i ; $i \in N$, $k \in K$
- L_{ik} : load for vehicle k leaving location i ; $i \in A$, $k \in K$
- l_{vk} , l_{ik} : load picked up/delivered by vehicle k from vendor v or at customer i ; $i, v \in N$, $k \in K$

The mixed integer programming model formulated for the presented problem is as follows:

$$\min \sum_{k \in K} \sum_{i \in A} \sum_{j \in A} c_{ijk} x_{ijk} + \sum_{k \in K} f_k y_k \tag{1}$$

$$\text{s.t. } \sum_{j \in N} x_{dj k} = y_k, \quad \forall k \in K \tag{2}$$

$$\sum_{j \in N} x_{j d k} = y_k, \quad \forall k \in K \tag{3}$$

$$\sum_{i \in A} x_{ijk} = \sum_{i \in A} x_{jik}, \quad \forall j \in A, \quad \forall k \in K \tag{4}$$

$$\sum_{k \in K} \sum_{v \in V} e_{vp} z_{ivk} \geq O_{ip}, \quad \forall i \in I_p, \quad \forall p \in P \tag{5}$$

$$\sum_{i \in I_p} z_{ivk} \leq M \sum_{j \in A} x_{jvk}, \quad \forall v \in V, \quad \forall k \in K \tag{6}$$

$$\sum_{k \in K} \sum_{j \in A} x_{jik} = 1, \quad \forall i \in I \tag{7}$$

$$T_{vk} + s_v + t_{vik} \leq T_{ik} + M(1 - z_{ivk}), \quad \forall i \in I_p, \quad \forall v \in V, \quad \forall k \in K \tag{8}$$

$$T_{ik} + s_i + t_{ijk} \leq T_{jk} + M(1 - x_{ijk}), \quad \forall i, j \in A, \quad \forall k \in K \tag{9}$$

$$a_i \leq T_{ik} \leq b_i, \quad \forall i \in A, \quad \forall k \in K \tag{10}$$

$$L_{dk} = \sum_{i \in I} \left(q_{id} \sum_{j \in A} x_{jik} \right), \quad \forall k \in K \tag{11}$$

$$l_{vk} = \sum_{i \in I_p} z_{ivk} q_{iv}, \quad \forall v \in V, \quad \forall k \in K \tag{12}$$

$$L_{jk} - l_{ik} \leq L_{ik} + M(1 - x_{jik}), \quad \forall i \in I, \quad \forall j \in A, \quad \forall k \in K \tag{13}$$

$$L_{jk} + l_{vk} \leq L_{vk} + M(1 - x_{jvk}), \quad \forall v \in V, \quad \forall j \in A, \quad \forall k \in K \tag{14}$$

$$L_{ik} \leq C_k, \quad \forall i \in A, \quad \forall k \in K \tag{15}$$

The objective function formulated in (1) requires the minimization of the variable and fixed costs of the distribution. The variable costs are defined by the road length traversed by the vehicles, whereas the fixed costs are based on the use of the vehicles. Constraints (2) and (3) ensure that the routes start and end at the depot, whenever the corresponding vehicle is used. Constraint (4) is used

for the vehicle flow balance. Constraint (5) identifies the pickup location for the ordered premium products. Constraint (6) is employed for consistency between the decision variables so that the routes are formed based on the customer and vendor stops of the vehicles. Constraint (7) ensures that each customer is visited only once. Constraint (8) provides that the premium products are picked up before the delivery. Constraint (9) ensures the time consistency throughout each route. Constraint (10) matches the delivery time with the time window of the customer. Constraint (11) provides that the initial load of the vehicle be equal to the customer orders from the depot. Constraint (12) identifies the total load to be picked-up at the vendor location. Constraints (13) and (14) ensure that the load quantity throughout the route is consistent with the customer delivery and vendor pickup quantities. Finally, constraint (15) is the vehicle capacity constraint.

Initially, we sought exact solutions for the formulated model using CPLEX. Such solutions within practical time periods have only been obtained for very small-sized problem instances. Such an instance with 10 customer nodes, 5 vendor nodes, 2 premium product types and 2 vehicles that is used to validate the model is depicted in Fig. 1. Loose depot due times compared to service times and loose vehicle capacities have been defined for this instance. The optimum solution of this instance is obtained as 784,362 cost units within a computational time period of 760.61 seconds.

We additionally tried larger-sized instances (with 25 customers, 75 vendors and 3 product types) in search for reasonable lower and upper bounds. After running CPLEX for 7200 seconds (2 hours), we were unable to obtain a lower or upper bound for any of the instances. Due to the complexity of the model implied in these early experiments, we have developed a hybrid metaheuristic solution methodology to solve realistically sized problems.

4. Solution methodology

The solution approach we have adopted for solving the CVRPMPTW described in the previous section is a hybrid metaheuristic approach. We employ two different heuristic algorithms for the vendor-assignment and routing aspects of our problem. We use a Genetic Algorithm (GA) for fixing the status of the vendor used for the assignment decisions, i.e. which vendor should be used for serving which (premium) customer order. The status of the vendor is defined as the vendor being “active” or “inactive”. So, only the vendors whose status is defined as active by the GA are used for the vendor–customer assignments. The CVRPMPTW problem is solved by a modified savings algorithm that we have

designed and implemented in this study. For a given vendor–customer assignment vector, the savings algorithm calculates the CVRPMPTW cost and this cost value is returned to GA in order to simultaneously find the best route and vendor–customer assignment. The overall structure of the hybrid metaheuristics solution approach is depicted in Fig. 2.

4.1. Genetic algorithm for assignment decisions

Genetic Algorithms are heuristic procedures inspired by biological and evolutionary principles and have been applied to optimization problems extensively. The main concept is to maintain a pool of solutions for the problem, usually referred to as a *population of chromosomes*, and continuously improve it through generations of *crossovers*. Through this process, the genetic material is carried over to offspring and the resulting solutions that are hopefully better in terms of objective function value, or the *fitness* value, are used to replace inferior solutions in the pool (Bozkaya et al., 2010). For further details on Genetic Algorithms, the interested reader is referred to the study by Reeves (2003).

In our approach, we have implemented GA using the C++ programming language because C++ is an appropriate language for dealing with computationally intensive functions and sub-routines. The fitness value of each individual in the population is defined as the CVRPMPTW cost, which is obtained from the modified savings algorithm implemented in Visual Basic for Applications (VBA). The modified savings algorithm has been coded in VBA, which is the programming platform of the ArcGIS software. ArcGIS is a commercial off-the-shelf software application for creating, storing, visualizing and analyzing spatial datasets and solutions. Thus, we aim to integrate the proposed algorithm with the ArcGIS environment. The VBA routine also maintains the data transfer between the two components of the solution methodology namely the GA and the modified savings algorithm.

In our adaptation, we have defined the chromosomes as each gene representing the customers and the encoding is made numerically where each number represents a vendor location. We use a constant-sized population, which is a fairly common feature in the facility location GA literature. That is, we take the size of the solution pool S as constant. Each chromosome in the population initially is generated in a random fashion. Two parent chromosomes are needed to produce an offspring. We choose the first parent randomly, and the second parent according to a gene diversity measure that we use. This measure is simply the total number of genes that are different in both parents. We use a traditional GA

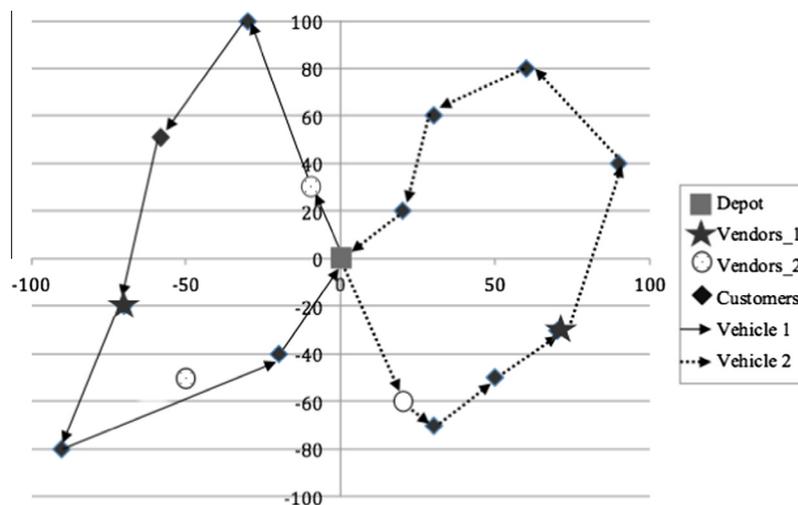


Fig. 1. Small instance with 10 customers, 5 vendors and 2 product types.

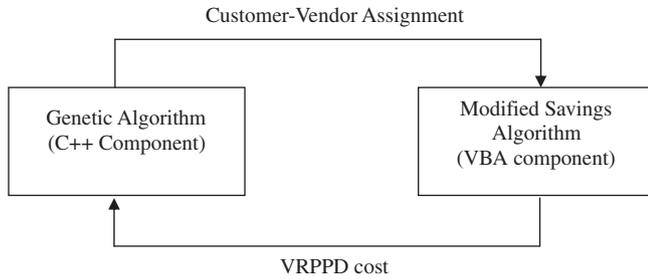


Fig. 2. Framework for the overall hybrid metaheuristic solution methodology.

crossover operator where a common crossover point on each parent chromosome is chosen and the portions of the two chromosomes after the crossover point are swapped. Our replacement strategy takes into consideration the offspring produced from two parents and uses it to replace the worst member of the pool. We keep track of the worst fitness value and the associated chromosome in the pool, and replace it with the new offspring only if the offspring fitness value is better. If not, the offspring is discarded and the algorithm continues to produce a new offspring. The GA algorithm generates a single offspring in each generation (or iteration) and terminates when it reaches the maximum number of generations. When the algorithm terminates, the best solution in the pool along with its objective function value is reported.

4.2. Modified savings algorithm for CVRPPD/TW

Our modified saving algorithm is implemented in a two-stage fashion where the routes are first constructed and then they are improved over an improvement search space. The following sections describe the main components of the proposed modified savings algorithm for CVRPPD/TW.

4.2.1. Construction phase

The construction phase is initiated by constructing a separate route for each customer including its assigned vendors if the customer has ordered premium goods. The vendors are assigned based on the minimum distance criterion and the status of the vendor (active or inactive) retrieved from the GA. Then, these routes are iteratively merged based on savings obtained, much similar to the Clarke-Wright savings algorithm. While calculating the cost saving of each merger, the following procedures are performed (during temporary mergers):

- *Elimination of extra vendors:* Since two routes are merged to construct a new route, these two routes may consist of two vendors that serve the same type of product. Only one of these vendors is required to supply the premium orders for the corresponding customers in the new route. Thus, the unnecessary vendors are eliminated.
- *Feasibility with respect to vehicle capacity and time windows:* The customers have time windows for accepting deliveries and the vehicles have due times to return to the depot. Thus, a merger may violate one or more customer time windows and the vehicle capacity. A feasibility check is performed to ensure such a violation is not the case.

If the merged routes are determined to be feasible according to the feasibility check, the cost savings of the routes are calculated. Cost savings are computed with two components: distance saving and time efficiency. The distance saving is calculated as the difference in the total of length of the two routes before and after the merger, which is then transformed into cost using a parameter. The time efficiency is evaluated based on minimizing the idle time

of the route after merger, which is also transformed into cost using a parameter. The overall cost saving of a candidate route is calculated using (16) where ρ is distance-to-cost parameter and τ is time-to-cost parameter,

$$\text{costsaving}(r_{\text{after_merger}}) = \Delta \text{dist}(r_i, r_j, \text{candidate_route}) \times (\rho + \text{TimeEff}(\text{candidate_route}, \text{TW}) \times \tau) \quad (16)$$

Once the savings from the merger of each pair is calculated, the merger with the highest saving is executed. The pseudo-code of the constructive heuristic is given in Algorithm 1.

Algorithm 1. Construction Heuristic

Input: G, I, V, K, α – status of vendors, $q, C, a_i, b_i, e, \rho, \tau$, and $IterLim$ – iteration limit

Output: *solution* – constructive initial solution

```

1: distlink = Do DistLink( $G, I, V$ )
2:  $\alpha$  = Call GeneticAlgorithm( $V$ )
3: assignment = Do Assignment(distlink,  $I, V, q, e, \alpha$ )
4: for  $i = 1$  to  $|I|$ 
5:    $r_i$  = Do Route( $I, \text{assignment}$ )
6: next  $i$ 
7: solution = Do Solution( $r$ )
8: costmatrix = Do CostMatrix(solution,  $\rho, \tau$ )
9: for iteration = 1 to IterLim
10:  for  $i = 1$  to  $|solution|$ 
11:   for  $j = i + 1$  to  $|solution|$ 
12:    candidate_route = Do TemporaryMerge( $r_i, r_j$ )
    where  $r_i, r_j \in solution$ 
13:    eliminate = Do Eliminate( $p_k, p_l, e, \text{costmatrix}$ )
     $\forall p_k, p_l \in \text{candidate\_route}(I, V), e(p_k) = e(p_l)$ 
14:    if IsFeasible(eliminate,  $C, a, b$ ) = True then
15:      costsaving(eliminate) =  $\Delta$ 
       $\text{dist}(r_i, r_j, \text{eliminate}) \times (\rho + \text{TimeEff}(\text{eliminate}, a, b) \times \tau)$ 
16:    end if
17:  next  $j$ 
18: next  $i$ 
19: savingmatrix = Do SavingMatrix(costsaving)
20: max_saving =  $\max(\text{costsaving}(r_i, r_j) \in \text{savingmatrix})$ 
21: new_route = Do Merge( $r_i, r_j$ )
22: solution = Do Solution(new_route,  $r_i, r_j$ )
23: next iteration
24: return solution =  $\{r_1, r_2, \dots, r_c\}$ 
  
```

4.2.2. Local search phase

Once the construction stage is completed, the routes are improved in an iterative fashion. In each iteration, the savings from three types of improvement moves are calculated. These are:

- Intra-route moves:* changing the sequence within routes.
- Inter-route moves:* swapping portions of two routes between each other.
- Replacements:* changing vendor locations with other vendors that supply the same type of product.

In each iteration, the best saving is calculated from all possibilities arising from these three types of moves and the corresponding move is implemented. The modified savings algorithm is illustrated in Fig. 3.

While searching for improvements with intra-route moves, an important feasibility check is performed to honor the sequence of

customers and the vendors used to fulfill the orders of these customers. The vendors should be visited before the customer who has orders to be supplied from these vendors. Thus, when the position of a vendor is being changed by an intra-route move, the vendor can only be placed before the customer it serves. Similarly, when a customer's position in the route is changed, it cannot be placed before a vendor serving this customer. In addition, feasibility with respect to vehicle capacity and time windows are also verified before completing an intra-route move. Subsequently, the cost saving of the move is calculated using (16). The pseudo-code of the intra-route move is presented in Algorithm 2.

The inter-route movements are more complex when compared to the intra-route moves due to customer–vendor assignments. Let us refer to the two routes subject to an inter-route move as *origin* route and *destination* route.

Algorithm 2. Intra-route Move

```

Input:  $I, V, \alpha, q, C, a_i, b_i, e, \rho, \tau, prev\_solution$  – solution from previous iteration or Algorithm 1
Output:  $max\_saving1$  – maximum saving with intra-route moves
1:  $solution = prev\_solution$ 
2:  $costmatrix = Do\ CostMatrix(solution, \rho, \tau)$ 
3: for  $r = 1$  to  $|solution|$ 
4:    $r' = r$ 
5:   if  $IsFeasible\_Sequence(r, k, l) = True$  then
6:      $r' = Do\ IntraMove(r, k, l) \forall k, l \leq |r|, k \neq l$  are two customer positions on route  $r$ 
7:   end if
8:    $r' = Do\ Eliminate(p_m, p_n, e, costmatrix)$  where  $p_m, p_n \in r'(I, V), e(p_m) = e(p_n)$ 
9:   if  $IsFeasible(r', C, a, b) = True$  then
10:     $costsaving1(r, k, l) = \Delta dist(r, r') \times (\rho + TimeEff(r', a, b) \times \tau)$ 
11:   end if
12: next  $r$ 
13:  $savingmatrix1 = Do\ SavingMatrix(costsaving1)$ 
14:  $max\_saving1 = \max(costsaving1(r, k, l) \in savingmatrix1)$ 
    
```

An inter-route move is initiated only if the location to be moved in an *origin* route is a customer location. As the customer to be swapped between routes is fixed, logical checks (as depicted in Fig. 4) are required in order to decide which locations will be swapped between the two routes due to customer–vendor assignments.

Similar to the intra-route moves, the position of the customer to be inserted in the destination route also needs to be decided after a *feasibility* check so as to maintain the correct sequence of customers and the vendors that serve them. If the additional vendors are inserted due to the moved customer, vendors are added just before the moved customer. Feasibility with respect to vehicle capacity and time windows is also verified during inter-route moves. Subsequently, the cost saving is calculated using (16), where the distance difference is between the total distances of the two routes before and after the move. The pseudo-code of the inter-route move is presented in Algorithm 3.

Algorithm 3. Inter-route Move

```

Input:  $I, V, \alpha, q, C, a_i, b_i, e, \rho, \tau, prev\_solution$  – solution from previous iteration or Algorithm 1
Output:  $max\_saving2$  – maximum saving with inter-route moves
1:  $solution = LocalSearch(prev\_solution, 1)$ 
2:  $assignment = LocalSearch(prev\_solution, 2)$ 
3:  $costmatrix = Do\ CostMatrix(solution, \rho, \tau)$ 
4: for  $r1 = 1$  to  $|solution|$ 
5:   for  $r2 = 1$  to  $|solution|$ 
6:     if  $IsVendorAssigned(assignment, r1, k) = False$ , where  $r1(k) \in I, k \leq |r1|$  then
7:       go to line12
8:     else if  $IsCustomerAssigned(assignment, r1, vendor(k)) = False$ , with  $r1 \setminus \{k\}$  then
9:        $r1 = r1 \setminus vendor(k)$ 
10:    end if
11:    if  $IsCustomerAssigned(assignment, r1, vendor(k)) = True$  then
12:      if  $IsFeasible\_Sequence(r2, k, l) = True$  then
13:         $i = Do\ InterMove(r1, k, l)$  where  $k \leq |r1|$ 
14:         $j = Do\ InterMove(r2, k, l)$  where  $k \leq |r1|, l \leq |r2|, k \neq l$ 
15:      end if
16:    else
17:       $r1 = Do\ InterMove(r1, k, l)$  where  $k \leq |r1|$ 
18:       $r2 = Do\ InterMove(r2, k, l)$  where  $k \leq |r1|, l \leq |r2|, k \neq l$ 
19:       $r2 = r2 \cup vendor(k)$ 
20:    end if
21:     $eliminate = Do\ Eliminate(r1, p_m, p_n, e, costmatrix)$  where  $p_m, p_n \in r1(I, V), e(p_m) = e(p_n)$ 
22:     $eliminate = Do\ Eliminate(r2, p_m, p_n, e, costmatrix)$  where  $p_m, p_n \in r2(I, V), e(p_m) = e(p_n)$ 
23:    if  $IsFeasible(r1, C, a, b) = True$  and  $IsFeasible(r2, C, a, b) = True$  then
24:       $costsaving2(r1, r2, k, l) = \Delta dist(r1_{beforemove}, r1_{aftermove}, r2_{beforemove}, r2_{aftermove}) \times (\rho + TimeEff(r1, r2, a, b) \times \tau)$ 
25:    end if
26:  next  $r2$ 
27: next  $r1$ 
28:  $savingmatrix2 = Do\ SavingMatrix(costsaving2)$ 
29:  $max\_saving2 = \max(costsaving2(r1, r2, k, l) \in savingmatrix2)$ 
    
```

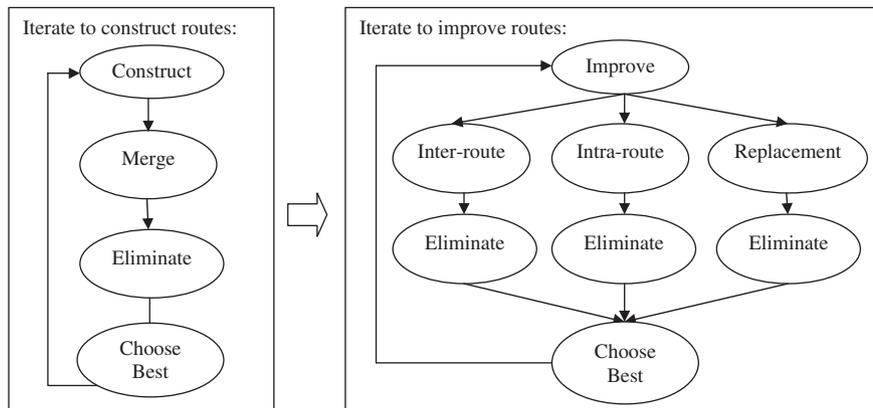


Fig. 3. Framework for the proposed modified savings algorithm methodology.

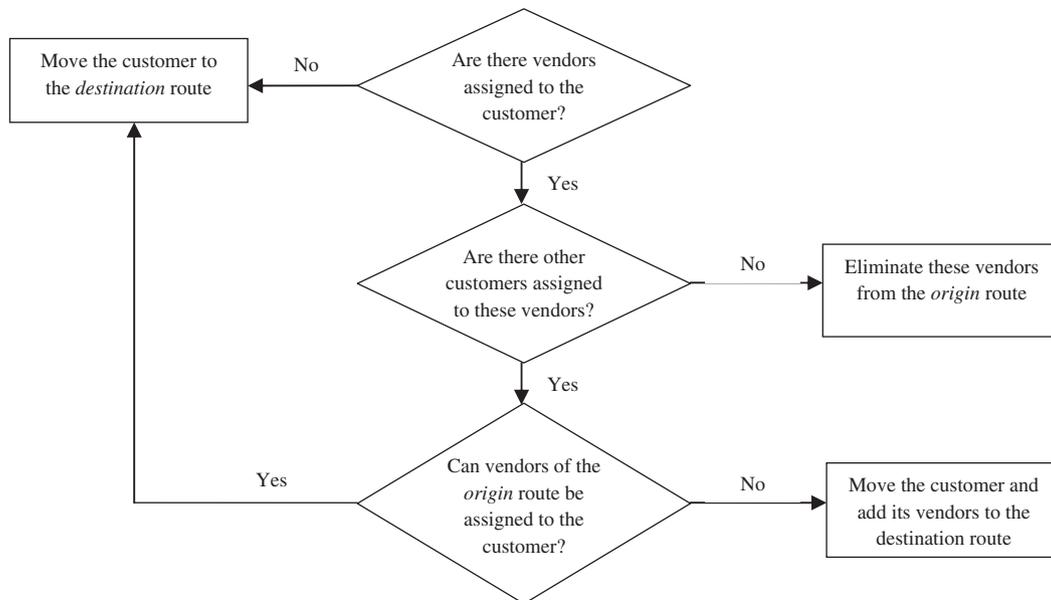


Fig. 4. Checks related to customer–vendor assignments during inter-route moves.

Additionally, in each iteration, after implementing an intra- or inter-route move, a vendor replacement search is conducted to possibly replace vendors of a route with better ones in the vendor list. The pseudo-code of the overall local search procedure is presented in Algorithm 4.

Algorithm 4. Local Search

Input: $V, \alpha, q, C, a_i, b_i, e, \rho, \tau, IterLim$
Output: $solution^{best}$ – the best overall solution

- 1: $(assignment, solution) = ConstructiveHeuristic()$
- 2: **for** $iteration = 1$ **to** $IterLim$
- 3: $max_saving1 = Intra-routeMoveHeuristic()$
- 4: $max_saving2 = Inter-routeMoveHeuristic()$
- 5: **if** $max_saving1 > max_saving2$ **then**
- 6: $new_route = Do\ IntraMove()$
- 7: $new_solution = Do\ Solution(new_route)$
- 8: **else**
- 9: $(new_route1, new_route2) = Do\ InterMove()$
- 10: $new_solution = Do\ Solution(new_route1, new_route2)$
- 11: **end if**
- 12: $assignment = Do\ Assignment(new_solution)$
- 13: $costmatrix = Do\ CostMatrix(new_solution)$
- 14: **for** $r = 1$ **to** $|new_solution|$
- 15: $new_route = r$
- 16: **for** $j = 1$ **to** $|r|$
- 17: **if** $IsVendor(j) = True$ **then**
- 18: $min_distance = Do\ Distlink(r)$
- 19: $new_supplier = j$
- 20: **for** $v = 1$ **to** $|V|$
- 21: $new_route = Do\ Replacement(j, v)$, where $\alpha(v) \neq 0$,
 $e(j) = e(v)$
- 22: **if** $distance(new_route) < distance(r)$ **and**
 $IsFeasible(new_route, C, a, b) = True$ **then**
- 23: $min_distance = DoDistlink(new_route)$
- 24: $new_supplier = v$
- 25: **end if**
- 26: **next** v

- 27: $new_route = Do\ Replacement(j, new_supplier)$
- 28: **end if**
- 29: **next** j
- 30: $new_solution = Do\ Solution(new_route)$
- 31: **next** r
- 32: $solution = new_solution$
- 33: **next** $iteration$
- 34: $solution^{best} = solution$

5. Computational results

In order to test the performance of our proposed algorithm, we first solve a set of Solomon benchmark problems. This widely known class of test problems is frequently used for comparative purposes in the literature. We then solve a real-world case, which is obtained from the geographical region of Istanbul, Turkey, to evaluate the potential profitability of the business model presented in this paper.

The testing of our proposed algorithm is conducted in two steps. Since there are no published instances and associated best-known solutions for the CVRPMPDTW, we first solve a set of published capacitated VRPTW instances from the Solomon instance library. Even though our algorithm is not particularly tailored to solve VRPTW, a sufficiently good performance with these instances should suggest that our algorithm could also effectively solve the CVRPMPDTW. Thus we adapt our overall solution methodology to solve capacitated VRPTW by eliminating the genetic algorithm procedure that is only required for customer–vendor assignments. We report the average results and gaps of each Solomon instance class in comparison to the average best heuristic results in the literature (Solomon, 2005) in Table 1. Results indicate that the average gap between the objective values of best-reported heuristic solutions of the Solomon instances and those of our solutions is 7.05%, which we find is reasonable for an algorithm that is not particularly tailored to solve VRPTW.

The Solomon instances listed in Table 1 all include 100 nodes as customer points and their delivery quantities, and are specified by the following characteristics:

Table 1

The comparison of the results for Solomon's 100-customer benchmark instances.

Instance class	Best known heuristic		Proposed algorithm				
	Average distance	Average iteration (construction)	Average distance (construction)	Average iteration (local search)	Average distance after local search	Average gap w.r.t. best known heuristic (%)	Std. dev. gap w.r.t. best known heuristic (%)
C1XX	828.31	92	951.29	6	867.67	4.76	4.27
C2XX	589.86	99	713.76	9	651.07	10.38	5.33
R1XX	1265.71	92	1663.87	47	1363.98	7.94	4.10
R2XX	966.18	100	1574.04	40	1062.46	9.57	6.04
RC1XX	1384.16	92	1670.75	28	1475.20	7.32	5.88
RC2XX	1119.35	100	1693.39	32	1186.01	5.98	2.91

- The placement of the customers is randomly distributed (R1XX and R2XX), clustered (C1XX and C2XX) or randomized clustered (RC1XX and RC2XX)
- Vehicle capacities are tight or loose (e.g. vehicle capacity of R1XX is 200 units but vehicle capacity of R2XX is 1000 units)
- Depot due times are tight or loose compared to service times (e.g. the service times of R1XX and R2XX are 10 with a depot due date of 230 for R1XX and 1000 for R2XX).

In our computational study, 8 instances for each of the above classes have been solved on an Intel Core 2 Quad CPU Q9400 2.67 gigahertz with a memory of 4 gigabyte RAM. The computation time of tight vehicle capacity instances is around 1–2 minutes. As the vehicle capacity increases, so does the computational complexity; thus, the time increases to approximately 10 minutes for instances with loose vehicle capacities.

In the second step of our computational study, we use our algorithm to solve modified Solomon problem instances that correspond to the CVRMPDPTW. The original Solomon problems consist of 100 customer delivery locations. We have modified these locations to randomly divide them into two groups: 25 customer delivery points and 75 potential vendor locations for three different types of premium products. We report the solutions of these instances computed by our algorithm in Table 2.

Note that the solution of the CVRMPDPTW does not necessarily visit each and every vendor pickup location, which means the total number of stops in all routes is not 100, as in the original Solomon instance solutions. Moreover, the solutions and the associated objective values reported in Table 2 correspond to different problems, namely the VRPTW and the CVRMPDPTW. Thus, it is not fair to compare these results with the published results for Solomon VRPTW instances even though we report them in the same table. We simply report our solutions as the first and best set of solutions for the new problem CVRMPDPTW we have introduced to literature.

The resulting routes for the Solomon CVRMPDPTW version are visualized in ArcGIS 9.3 software and depicted in Fig. 5. The depot, which is the start and end point for each route is shown with the green square in the middle of the figure. The customers are represented by circles and the vendors are represented by triangles in the figure.

6. Case study

We now test our solution methodology with a sample dataset for the city of Istanbul. In this case, we use two scenarios to show the potential profit increase that can be obtained with the business model presented in this paper. In the first scenario, an e-supermarket delivers grocery products to several locations in Istanbul in the classical sense, i.e., using a fleet that originates at the supermarket and delivers products only available in the supermarket. In the second scenario, the e-supermarket utilizes an extended fleet to pick-up premium products from multiple vendor locations and delivers them consolidated with the standard grocery products to the customer locations.

The study area covers part of the European side of Istanbul, specifically the Besiktas, Sisli and Sariyer districts, located in a densely populated part of the city center with usual traffic difficulties. We consider four different premium products, namely organic goods, specialty goods, books and sports goods. In Fig. 6, these are marked with the flower, star, shield and triangle, respectively. Also in Fig. 6 are the customer locations (check symbol) and the e-delivery store (depot) location (black plus symbol).

In the first (baseline) scenario, we solve the traditional e-delivery routing problem with no premium goods. In this problem, a fixed number of trucks leave the store with all standard products loaded and visit customers sequentially to make deliveries. Based on a prior study (deKervenoael et al., 2011), where we conducted a survey with e-supermarket industry leaders and determined parameters related to delivery operations and e-grocery sales, we assume that 50 customers place orders for e-delivery with an average basket size of 100 Turkish Lira (1 USD = 1.8 Turkish Lira) and an overall bottom line profit margin of 5%, within the regional and time span of the e-delivery activity.

In the second scenario, when additional premium products are available for purchase, we expect that some existing customers will extend their current orders with the added choices. The additional profit generated will (hopefully) offset the increased logistics costs in the CVRMPDPTW version. In our example, a total of 50 customers requests e-delivery with an average basket size of 120 Turkish Lira. Out of this 120 Turkish Lira, the average profit achieved is increased to 10% with the additional orders for premium products. We present in Fig. 7a and b, the routing solutions

Table 2

The results with modified Solomon instances for CVRMPDPTW.

Instance	Proposed algorithm			
	Total nr. of stops	Nr. of vehicles	Total cost	Computation time (seconds)
C101	52	7	702.0	1200
C201	36	3	511.2	1140
R101	46	10	897.0	960
R201	33	2	541.2	1860
RC101	50	11	960.0	1800
RC201	34	2	669.8	1680

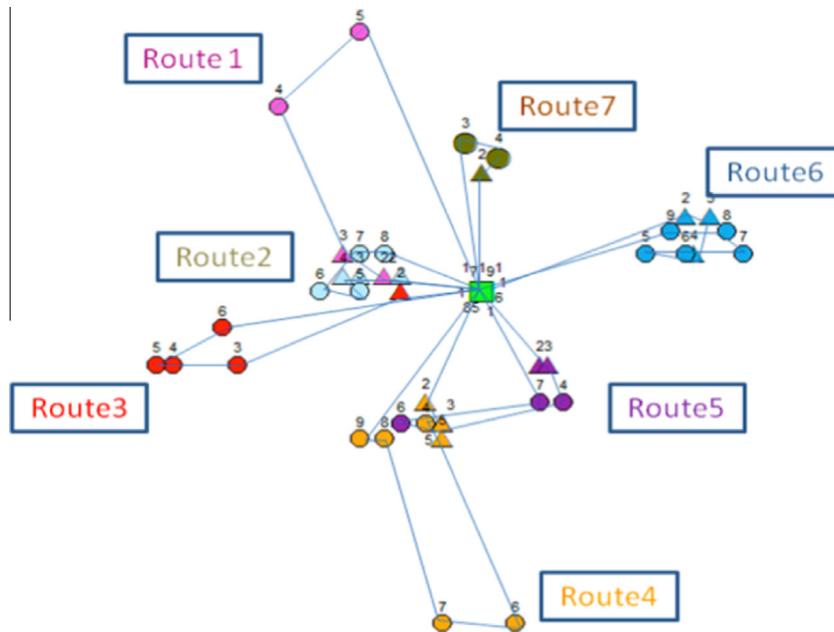


Fig. 5. The resulting routes for a modified Solomon CVRMPDWTW instance.

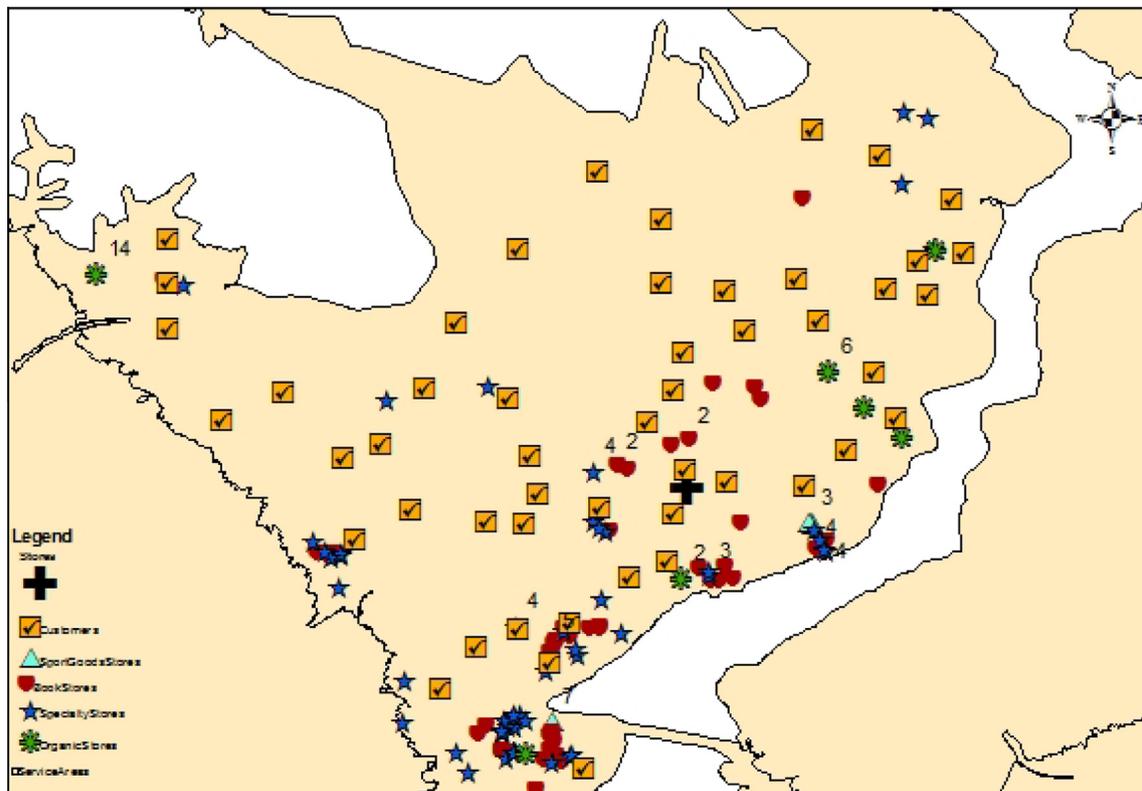
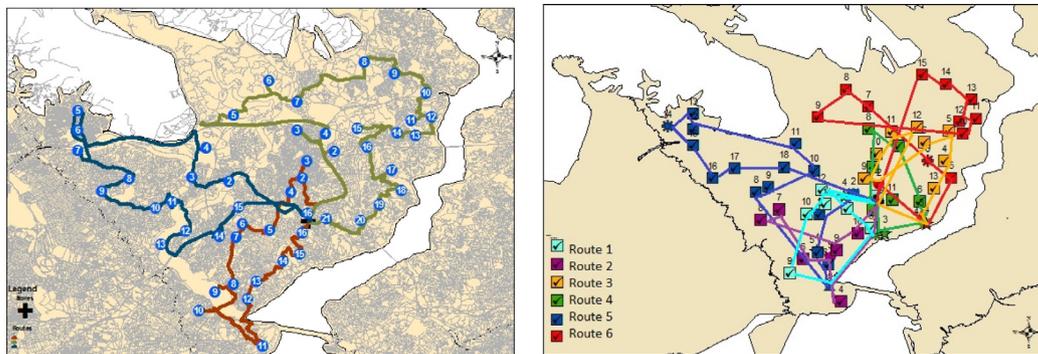


Fig. 6. Problem setting in Istanbul.

obtained by running the traditional VRP model and the CVRMPDWTW model.

In the case study, e-deliveries are grouped into specific time slots based on the time window preferences of customers. Thus, the deliveries in the same time slot may be treated as deliveries without time windows. So, we treat the case study problem as if there is no time window, even though the algorithm we have developed for CVRMPDWTW is able to honor the time windows

when specified. Once we obtain the results in terms of distance traversed for the first and second scenarios, we convert the distance into distribution costs using a specific parameter based on the oil consumption rate and oil prices (0.35 Turkish Lira/kilometer). We also consider the total fixed cost of a vehicle including fixed vehicle costs such as insurance, driver salary, etc. We assume 2500 Turkish Lira/month salary for drivers with monthly total working hours of 225. We calculate the fixed cost of a route by converting the



	(a)	(b)
Nr. of Customers Visited	50	50
Average Basket Size (TL)	100	120
Average Margin	5%	10%
Used Nr of Vehicles	3	6
Nr of Stops Visited	50	73
Distance Traversed (km)	122.10	217.50
Distribution Cost (TL)	42.70	75.95
Fixed Cost (TL)	67.83	120.83
Net Profit (TL)	139.47	403.22

Fig. 7. Results for the case study comparing classical VRP vs. CVRPMPDTW.

Table 3 Additional profit levels (in Turkish Lira) with respect to changes in the average basket size and average profit margins.

The increase in the basket size (Turkish Lira)	Average margin					
	5%	6%	7%	8%	9%	10%
-30	-161.25	-126.25	-91.25	-56.25	-21.25	13.75
-20	-136.25	-96.25	-56.25	-16.25	23.75	63.75
-10	-111.25	-66.25	-21.25	23.75	68.75	113.75
+0	-86.25	-36.25	13.75	63.75	113.75	163.75
+10	-61.25	-6.25	48.75	103.75	158.75	213.75
+20	-36.25	23.75	83.75	143.75	203.75	263.75

Table 4 Profit increase in response to an increase in the number of customers.

	VRP	CVRPMPDTW with increase in margin and basket size	CVRPMPDTW with increase in margin and basket size and customers
Nr. of customers visited	50	50	60
Average basket size (Turkish Lira)	100	120	120
Average margin	5%	10%	10%
Used nr. of vehicles	3	6	6
Nr. of stops visited	50	73	80
Distance traversed (kilometer)	122.10	217.50	272.0
Distribution cost (Turkish Lira)	42.70	75.95	95.20
Fixed cost (Turkish Lira)	67.83	120.83	151.11
Net profit (Turkish Lira)	139.47	403.22	473.69
Profit increase (Turkish Lira)		+263.75	+334.22

distance to time using an average speed of 20 kilometer/hour and then multiplying the time with the total fixed cost per time. To obtain the solutions reported and illustrated in Fig. 7, we assume a specific increase in the basket size and average profit margin of the basket due to the introduction of premium goods.

6.1. Sensitivity Analysis

The increase in the basket size and profit margin clearly may not be foreseen in advance. In order to evaluate the profitability

of introducing premium products in a CVRPMPDTW, we present a sensitivity analysis in Table 3. The values in the first column indicate how much the basket size is changed in the sensitivity analysis in relation to the base scenario value of 100 Turkish Lira in Fig. 7a. The values in the table indicate the additional profit (or loss) obtained with respect to the value of 139.47 Turkish Lira. The analysis shows that the costs incurred by additional distance traversed in the extended operations of CVRPMPDTW; hence, the increased transportation cost can be offset only when the basket size and average margin are at a certain level. For example, when

Table 5

Additional profit levels (in Turkish Lira) with respect to changes in the average basket size and average profit margins with the increased number of customers.

The increase in the basket size (Turkish Lira)	Average margin					
	5%	6%	7%	8%	9%	10%
–30	–175.78	–133.78	–91.78	–49.78	–7.78	34.22
–20	–145.78	–97.78	–49.78	–1.78	46.22	94.22
–10	–115.78	–61.78	–7.78	46.22	100.22	154.22
+0	–85.78	–25.78	34.22	94.22	154.22	214.22
+10	–55.78	10.22	76.22	142.22	208.22	274.22
+20	–25.78	46.22	118.22	190.22	262.22	334.22

the basket size decreases by 30 Turkish Lira, the new operational setting defined with pickup and deliveries is relatively more profitable only with a margin of at least 10%.

On the other hand, one may further consider that new customers may place orders when additional premium products are available for purchase. Such an increase in the number of customers will also increase the level of profits as illustrated in Table 4.

A similar sensitivity analysis for CVRPMPDTW has been conducted with the increased number of customers as well as varying margins and value of basket size. The results in Table 5 show that under the new setting, where premium products lead to increase in the number of customers, the new operational setting with longer routes and more vehicles potentially offers more profits (or reduced loss) for most of the parameter combinations.

7. Conclusion

The comparison of the results obtained with the proposed solution methodology in this paper and the solution of Solomon problem stated in the literature show that our results are indeed promising for a new problem and a new solution approach developed. Since the CVRPMPDTW is subject to a higher number of constraints such as vendor–supplier sequence and the changing load level throughout the route–vehicle capacity, it is expected that the resulting cost of the CVRPMPDTW would be higher than the classical VRP.

On the other hand, our case study analysis indicates that there are forgone profit opportunities for e-supermarkets due to our proposed business model. As shown in illustrative scenarios, involvement of premium goods offers possibilities for improving profits conditional to an increase in the number of customers served, the value of the basket size and/or margin.

Future research may be extended on the concept of collaboration between the e-grocery and the vendors in this study to the collaboration among multiple chains of e-groceries and vendors. Thus, orders of customers from different e-supermarkets may be picked up and delivered in a unified operation. This problem will yield a multi-depot pickup and delivery type model illustrated in this study. Further research may also consider extending the model to include a reverse logistics dimension of the operation, namely collecting returned or unwanted goods on the way back to the store or depot.

In another lane of research, the problem may be expanded by including the dynamic nature of the business. In this case, orders that arrive after routing is completed may be added into the routing decision dynamically.

Acknowledgments

This research has been conducted in partial support of the TUBITAK research Grant #108K345. The authors are thankful to TUBITAK for this support.

The authors also thank Prof. Stephen Graves for his feedback during the revision phase of this manuscript, and Ali Yeşilçimen for his support during computational experiments with CPLEX.

References

- Aksen, D., Altinkemer, K., 2008. A location-routing problem for the conversion to the 'click-and-mortar' retailing: the static case. *European Journal of Operational Research* 186, 554–575.
- Attanasio, A., Cordeau, J.F., Ghianic, G., Laporte, G., 2004. Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. *Parallel Computing* 30 (3), 377–387.
- Balas, E., Fischetti, M., Pulleyblank, W.R., 1995. The precedence-constrained asymmetric traveling salesman polytope. *Mathematical Programming* 68 (1–3), 241–265.
- Baldacci, R., Hadjiconstantinou, E., Mingozzi, A., 2003. An exact algorithm for the traveling salesman problem with deliveries and collections. *Networks* 42 (1), 26–41.
- Bent, R., van Hentenryck, P., 2006. A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. *Computers & Operations Research* 33 (4), 875–893.
- Berbeglia, G., Cordeau, J.F., Gribkovskaia, I., Laporte, G., 2007. Static pickup and delivery problems, a classification scheme and survey. *TOP* 15 (1), 1–31.
- Bianco, L., Mingozzi, A., Ricciardelli, S., 1994. A set partitioning approach to the multiple depot vehicle scheduling problem. *Optimization Methods and Software* 3 (1–3), 163–194.
- Bodin, L.D., Sexton, T., 1986. The multi-vehicle subscriber dial-a-ride problem. *TIMS Studies in the Management Sciences* 22, 73–86.
- Borndörfer, R., Klostermeier, F., Grötschel, M., and Küttner, C., 1997. *Telebus Berlin, Vehicle Scheduling in a Dial-a-Ride System*. Technical Report SC 97–23, Konrad-Zuse-Zentrum für Informationstechnik, Berlin.
- Caricato, P., Ghiani, G., Grieco, A., Guerriero, E., 2003. Parallel tabu search for a pickup and delivery problem under track contention. *Parallel Computing* 29, 631–639.
- Bozkaya, B., Yanik, S., Balcisoy, S., 2010. A GIS-based optimization framework for competitive multi-facility location-routing problem. *Networks and Spatial Economics* 10 (3), 297–320.
- Cordeau, J.-F., Laporte, G., 2003a. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part B* 37, 579–594.
- Cordeau, J.-F., Laporte, G., 2003b. The dial-a-ride problem (DARP), variants, modeling issues and algorithms. *4OR, A Quarterly Journal of Operations Research* 1, 89–101.
- Cordeau, J.-F., Laporte, G., 2007. The dial-a-ride problem, models and algorithms. *Annals of Operations Research* 153, 29–46.
- Cordeau, J.-F., Laporte, G., Ropke, S., 2008. Recent models and algorithms for one-to-one pickup and delivery problems. In: Golden, B., Raghavan, S., Wasil, E. (Eds.), *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer US, vol. 43, pp. 327–357.
- Cullen, F.H., Jarvis, J.J., Ratliff, H.D., 1981. Set partitioning based heuristics for interactive routing. *Networks* 11 (2), 125–143.
- deKervenoael, R., Bozkaya, B., Palmer, M., 2011. Resisting collaborative practice: an analysis of supply chain partnerships for building urban online grocery provision. In: Malik, Kamna (Ed.), *Business Organizations and Collaborative Web-Practices, Strategies and Patterns*. IGI Global Publishing.
- Desaulniers, G., Desrosiers, J., Erdmann, A., Solomon, M.M., Soumis, F., 2002. VRP with pickup and delivery. In: Toth, P., Vigo, D. (Eds.), *The Vehicle Routing Problem*. SIAM, Philadelphia, pp. 225–237.
- Desrosiers, J., Dumas, Y., Soumis, F., Taillefer, S., Villeneuve, D., 1991. An algorithm for mini-clustering in handicapped transport. *Les Cahiers du GERAD, G-91-02*, HEC Montréal.
- Desrosiers, J., Dumas, Y., Soumis, F., 1986. A dynamic programming solution of the large-scale single vehicle dial-a-ride problem with time windows. *American Journal of Mathematical and Management Sciences* 6, 301–325.
- Doerner, K., Hartl, R.F., Reimann, M., 2001. Ants solve time constrained pickup and delivery problems with full truckloads. In: *Operations Research Proceedings*, pp. 395–400.

- Doerner, K., Hartl, R.F., Maniezzo, V., Reimann, M., 2003. An ant system metaheuristic for the capacitated arc routing problem. In: Proceedings of the 5th metaheuristics international conference, Tokyo, Japan.
- Dumas, Y., Desrosiers, J., Soumis, F., 1989. Large scale multi-vehicle dial-a-ride problems. *Les Cahiers du GERAD*, G-89-30, HEC Montréal.
- Fischetti, M., Toth, P., 1989. An additive bounding procedure for combinatorial optimization problems. *Operations Research* 37 (2), 319–328.
- Ganesh, K., Narendran, T.T., 2007. CLOVES: a cluster-and-search heuristic to solve the vehicle routing problem with delivery and pick-up. *European Journal of Operational Research* 178 (3), 699–717.
- Gendreau, M., Potvin, J.Y., Bräumlaysy, O., Hasle, G., Løkketangen, A., 2008. Metaheuristics for the vehicle routing problem and its extensions, a categorized bibliography. In: Golden, B., Raghavan, S., Wasil, E. (Eds.), *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer US, vol. 43, pp. 143–169.
- Hart, S.M., 1996. The modeling and solution of a class of dial-a-ride problems using simulated annealing. *Control and Cybernetics* 25 (1), 131–157.
- Hernandez-Perez, H., Salazar-Gonzalez, J.J., 2004. A branch- and cut algorithm for the traveling salesman problem with pickup and delivery. *Discrete Applied Mathematics* 145, 126–139.
- Hernández-Pérez, H., Salazar-Gonzalez, J.J., 2007. The one-commodity pickup-and-delivery traveling salesman problem: inequalities and algorithms. *Networks* 50 (4), 258–272.
- Jaw, J.J., Odoni, A.R., Psaraftis, H.N., Wilson, N.H.M., 1986. A heuristic algorithm for the multi-vehicle advance-request dial-a-ride problem with time windows. *Transportation Research Part B* 20 (3), 243–257.
- Kalantari, B., Hill, A.V., Arora, S.R., 1985. An algorithm for the traveling salesman problem with pickup and delivery customers. *European Journal of Operations Research* 22, 377–386.
- Landrieu, A., Mati, Y., Binder, Z., 2001. A tabu search heuristic for the single vehicle pickup and delivery problem with time windows. *Journal of Intelligent Manufacturing* 12, 497–508.
- Li, H., Lim, A., 2001. A metaheuristic for the pickup and delivery problem with time windows. In: 13th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'01). IEEE Computer Society, Los Alamitos, CA, pp. 333–340.
- Lin, C.K.Y., 2008. A cooperative strategy for a vehicle routing problem with pickup and delivery time Windows. *Computers and Industrial Engineering* 55, 766–782.
- Lu, Q., Dessouky, M.M., 2006. A new insertion-based construction heuristic for solving the pickup and delivery problem with time windows. *European Journal of Operations Research* 175, 672–687.
- Melachrinoudis, E., Ilhan, A.B., Min, H., 2007. A dial-a-ride problem for client transportation in a healthcare organization. *Computers & Operations Research* 34, 742–759.
- Montane, F.A., Galvao, R.D., 2006. A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers & Operations Research* 33 (3), 595–619.
- Nagy, G., Salhi, S., 2005. Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *European Journal of Operational Research* 162 (1), 126–141.
- Nanry, W.P., Barnes, W., 2000. Solving the pickup and delivery problem with time windows using reactive tabu search. *Transport Research Part B* 34, 107–121.
- Pankratz, G., 2005. A grouping genetic algorithm for the pickup and delivery problem with time windows. *OR Spectrum* 27, 21–41.
- Parragh, S.N., Doerner, K.F., Hartl, R.F., 2008a. A survey on pickup and delivery problems. Part I: Transportation between customers and depot. *Journal für Betriebswirtschaft* 58 (1), 21–51.
- Parragh, S.N., Doerner, K.F., Hartl, R.F., 2008b. A survey on pickup and delivery problems. Part II: Transportation between pickup and delivery locations. *Journal für Betriebswirtschaft* 58 (1), 81–117.
- Rekiek, B., Delchambre, A., Saleh, H.A., 2006. Handicapped person transportation, an application of the grouping genetic algorithm. *Engineering Application of Artificial Intelligence* 19, 511–520.
- Reeves, C., 2003. Genetic algorithms. In: Glover, F., Kochenberge, G.A. (Eds.), *Handbook of Metaheuristics*. Kluwer, Boston, pp. 55–82.
- Solomon, M.M., 2005. VRPTW Benchmark Instances, <http://w.cba.neu.edu/msolomon/problems.htm>. Last updated on March 24, 2005. Last accessed on May 30, 2013.
- Thangiah, S.R., Fergany, A., Awan, S., 2007. Real-time split-delivery pickup and delivery time window problems with transfers. *Central European Journal of Operations Research* 15 (4), 329–349.
- Toth, P., Vigo, D., 1996. Fast local search algorithms for the handicapped persons transportation problem. In: Osman, I.H., Kelly, J.P. (Eds.), *Metaheuristics: Theory and Applications*. Kluwer, Boston, pp. 677–690.
- Toth, P., Vigo, D., 1997. Heuristic algorithms for the handicapped persons transportation problem. *Transportation Science* 31, 60–71.
- Toth, P., Vigo, D., 2002. Models, relaxations and exact approaches for the capacitated vehicle routing problem. *Discrete Applied Mathematics* 123 (1–3), 487–512.
- Yanik, S., deKervenoael, R., Bozkaya, B., Palmer, M. and Koçoğlu, I., 2010. Discriminating behaviors through value added premium online grocery offering. In: 34th Annual Conference of the German Classification Society, Karlsruhe, July 21–23, 2010.
- Van Der Bruggen, J.J., Lenstra, J.K., Schuur, P.C., 1993. Variable-depth search for the single-vehicle pickup and delivery problem with time windows. *Transportation Science* 27 (3), 298–311.