Hybrid approach for the Multiple Vehicle Routing Problem with Deliveries and Selective Pickups

Bruno Petrato Bruck, André Gustavo dos Santos
Departamento de Informática
UFV - Universidade Federal de Viçsa
Viçosa, MG - Brazil
bruno.p.bruck@gmail.com, andre@dpi.ufv.br

***Abstract*—We propose the Multiple Vehicle Routing Problem with Delivery and Selective Pickups (MVRPDSP) along with a mixed integer linear programming formulation and a hybrid cluster-first heuristic. We show that it is possible to divide the cluster-first heuristic in sub problems which can be solved exactly and their solutions combined in a semi-greedy way to generate good solutions. The results show that while the model is not able to find even a feasible solution for several instances the heuristic finds, for most cases, better or equal solutions in a matter of milliseconds, including 4 optimal solutions, what proves its efficiency. We also tested a combined approach, using the heuristic to generate an initial solution for the model, which improved even more the results of the heuristic.**

***Keywords*-vehicle routing; combinatorial optimization; hybrid methods;**

I. Introduction

In the Multiple Vehicle Routing Problem with Deliveries and Selective Pickups (MVRPDSP) there are a set of customers to be served and a depot from where the vehicles depart to serve the customers. Each customer has a certain demand of goods either to be delivered or to be picked up, which generates a revenue if collected. It is possible for a customer to have both demands. In such case, they can be performed simultaneously or in two different visits, each completely fulfilling one of the demands. Each vehicle that departs from the depot shall perform a route that visits a subset of customers performing deliveries and pickups, then return to the depot. All delivery demands must be fulfilled exactly once by any of the available vehicles. The pickup demands are not mandatory, therefore they are only performed if there is enough space in the vehicle and if they are profitable, what implies that the revenue generated by collecting these pickups is greater than the additional routing cost. One can notice that some pickups might not be served at all by any vehicle and it is possible to argue that they would need to be performed at some point. To address this issue these pickups could either be added into another set of customers, like the set of customers that will be served in the following day, or a third party service can be used to collect these pickups, which could be a less costly option than forcing all pickups to be fulfilled or sending another vehicle only to perform few pickups. The objective is to find a set of routes that minimizes the total routing cost, which is the travel cost between the customers of each route minus the revenue generated by the collected pickups.

Figure 1 shows a small example with seven customers. In this example the route of vehicle 1 has a transportation cost of $5 + 4 + 4 + 10 = 23$ and it generates a revenue of 5 by performing one pickup demand, therefore the total cost of this route is $23 - 5 = 18$. The second route has a transportation cost of 20, but it generates a revenue of $8 + 20 = 28$, therefore the total cost of such route is -8, which means that this route ended up generating a profit greater than its cost. Considering both routes the total cost of this solution is $18+(-8) = 10$. Notice that route 2 cannot be done in the reverse order, since there is no space in the vehicle to collect the first pickup demand because in this case it would be still full with the delivery goods.

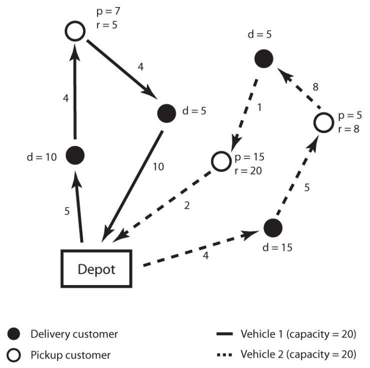


Figure 1. Example of a solution with 2 vehicles of capacity 20

This problem is an extension of the Single Vehicle Routing Problem with Deliveries and Selective Pickups (SVRPDSP) by adding the possibility of having multiple vehicles and consequently, multiple routes. Therefore, it has at least the same practical applications of the single vehicle version, arising in several reverse logistics contexts. An example is drink factories, which besides having to supply stores and supermarkets with full bottles, has to pickup empty bottles, returning them to the factory in order to be clean and refilled [1]. This problem also arises in logistic companies, which have delivery demands of packages and also pickup demands of goods that will be transported and delivered elsewhere. Another context is referred by [2] and [3], in which as a consequence of the constant growth in global awareness to preserve the environment, companies that manufactures electronics and batteries are being responsible to also pickup their broken and used products that otherwise would go as normal trash and pollute.

The MVRPDSP can be described using a complete directed graph $G = \{V, A\}$, where $V = \{0, 1, ..., n\}$ is a vertex set representing the depot (vertex 0) and the n customers and $A = (i, j) : i, j \in V, i \neq j$ is a set of arcs. Each customer i has a delivery demand $d_i \geq 0$, a pickup demand $p_i \geq 0$, and a revenue $r_i \geq 0$. Each arc $(i, j) \in A$ has a travel cost c_{ij}. Let K be the set of vehicles. We assume that all vehicles in K have the same capacity Q, therefore the fleet is homogeneous. This problem is clearly NP-hard since it can be reduced to a special case of the well known Traveling Salesman Problem (TSP) when $|K| = 1$, $Q \geq \sum_i^n d_i$ and $p_i = 0 \; \forall i \in V$.

The remainder of this paper is organized as follows: section II provides a review of the related literature; section III presents a mathematical formulation proposed to solve this problem; a hybrid constructive heuristic is described in section IV; section V presents the computational results for the proposed approaches including the results for the model using the constructive to provide an initial solution; finally section VI presents some conclusions and propose some further investigations.

II. Related Literature

To our knowledge, the present work is the first to address the MVRPDSP, but as this problem is an extension of the SVRPDSP, its literature review is very important and can provide some insights for the multiple vehicle version.

Sur̈al and Bookbinder [5] are the first to directly address the SVRPDSP. They present the problem using the notation $\alpha/\beta/\gamma$, where α denotes the number of vehicles (1 for single and M for multiple), β the pickup service options ($must$ or $free$ if the pickup is respectively mandatory or optional) and γ the precedence order for visiting two customers ($prec$ if all deliveries must precede the pickups, or any if they can be visited in any order). While the SVRPDSP is *1/free/any*, according to this notation, the MVRPDSP can be described as *M/free/any*. They cited papers dealing with the *1/free/prec* and *1/must/any* problems, and claimed to be the first to address the *1/free/any*. For the multiple vehicle versions, they list papers for the *M/must/prec* and *M/free/prec*, but no mention is made about the *M/free/any*, which is our object of study. They proposed a MILP formulation for the single vehicle version and some improvements in order to strengthen the formulation such as constraint disaggregation, coefficient improvement, cover and logical inequalities, and lifted subtour elimination constraints. They were able to solve to optimality about 75% of their test instances, which contained at most 30 customers.

In [6], another formulation is proposed for the single vehicle version, and also a Tabu Search (TS) comparing their performance with some classical constructive heuristics and a lower bound calculated based on the solution of the TSP (considering only the delivery customers) and the solution of the Knapsack Problem (considering only the pickups). They tested their approaches using a set of 68 instances derived from 17 instances of the Capacitated Vehicle Routing Problem (CVRP) obtained from the VRPLIB [7].

Another mention to the SVRPDSP is made by Laporte and Gribkovskaia in [4], in which they present the One-to-Many-to-One Single Vehicle Pickup and Delivery problem and cites the SVRPDSP as an extension that had received limited attention, despite its many practical application and similarity to other vehicle routing problems.

Coelho [7], in a master's thesis, proposed a General Variable Neighborhood (GVNS) to solve the SVRPDSP obtaining better results than the TS for the same instances. Recently a Hybrid Evolutionary Algorithm (EA) was proposed by [2] and the results compared to the previous approaches, yielding better results for some instances including 2 new optimal solutions. Bruck and Santos [3] report the results of some improvements on the EA and the constructive algorithms used in [2], greatly improving the previous results of the EA and outmatching the GVNS in more than half of the instances, including 7 optimal solutions. They also proposed a Variable Neighborhood Descent (VND) which, although was not as efficient as the EA, has found about 1/3 of solutions equal or better than the GVNS.

III. MILP formulation for the MVRPDSP

Our formulation is based on the one proposed by Süral and Bookbinder [5] for the SVRPDSP, which uses the Miller-Tucker-Zemlin (MTZ) subtour elimination constraints. We extended this formulation and its improvements to accept multiple vehicles and included two additional improvements in order to remove symmetry.

Let us define the following notation, which will be used from now on to describe our input data:

- V_d: set of delivery customers
- V_p: set of pickup customers
- V: set of nodes ($V = V_d \cup V_p \cup 0$)
- K: set of vehicles
- d_i: delivery demand of customer i
- p_i: pickup demand of customer i
- r_i: revenue associated with the pickup of customer i
- c_{ij}: cost for travelling arc (i, j)
- D: total delivery demand ($D = \sum_{i \in V_d} d_i$)
- P: total pickup demand ($P = \sum_{i \in V_p} p_i$)
- Q: vehicle capacity

Notice that this notation does not allow a customer to have both demands at the same time. Therefore, in such case, the customer is split into two different customers, one having the delivery demand, the other with the pickup demand, and assuming the transportation cost between the two as 0.

This formulation uses three sets of decision variables. Let $x_{ij}^k = 1$ if customer i immediately precedes customer j ($i \neq j$) in route k and 0 otherwise. As in [5], in order to define the subtour elimination constraints, we define y_i as the total delivery load of a vehicle when departing from customer i. Similarly, we define z_i as the total pickup load of a vehicle when arriving at customer i. Notice that there is no reference of which vehicle these variables are related to, this is due to the fact that at most one vehicle will visit a given customer, therefore there is no need to directly relate these variables to each vehicle. Then our mathematical model for the MVRPDSP is as the following:

$$Min \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij}^k - \sum_{k \in K} \sum_{i \in V} \sum_{j \in V_p} r_j x_{ij}^k \tag{1}$$

Subject to:

$$\sum_{k \in K} \sum_{i \in V} x_{ij}^k = 1 \qquad \forall j \in V_d \tag{2}$$

$$\sum_{i \in V} \sum_{k \in K} x_{ij}^k \leq 1 \qquad \forall j \in V_p \tag{3}$$

$$\sum_{j \in V \setminus \{0\}} x_{0j}^k \leq 1 \qquad \forall k \in K \tag{4}$$

$$\sum_{j \in V} x_{ij}^k - \sum_{j \in V} x_{ji}^k = 0 \qquad \forall i \in V, k \in K \tag{5}$$

$$y_i - y_j \geq d_j - D(1 - \sum_{k \in K} x_{ij}^k) \qquad \forall i, j \in V \setminus \{0\} \tag{6}$$

$$z_j - z_i \geq p_i - Q(1 - \sum_{k \in K} x_{ij}^k) \qquad \forall i, j \in V \setminus \{0\} \tag{7}$$

$$(y_i + d_i) + (z_i + p_i) \leq Q \qquad \forall i \in V \setminus \{0\} \tag{8}$$

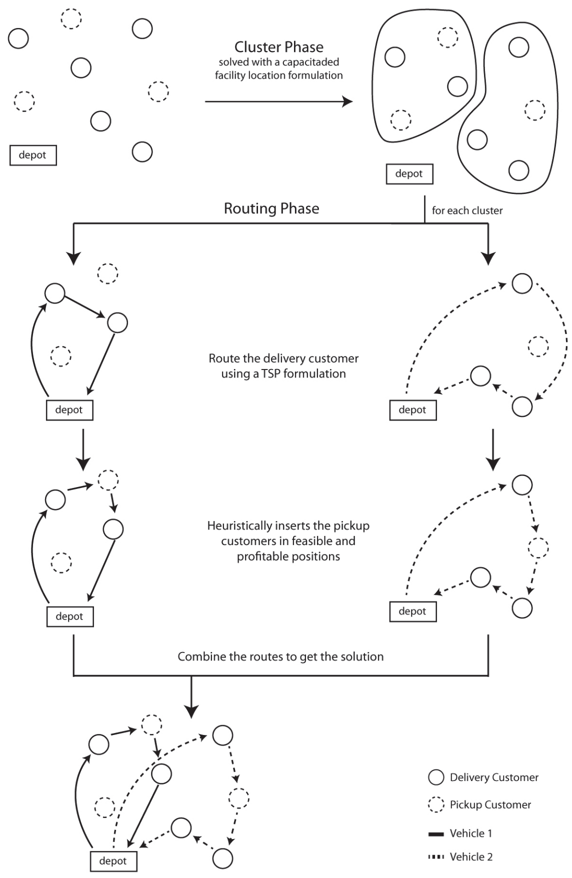
$$y_i, z_i \geq 0, x_{ij}^k = \{0, 1\} \qquad \forall i, j \in V, k \in K \tag{9}$$

The objective is to minimize the function (1), that represents the total cost of the solution, which is the sum of the routing cost of each route minus the revenue generated by all pickups collected. Constraints (2) specify that all delivery customers must be visited exactly once and by only one vehicle. The set of constraints (3) define that pickup customers can be visited at most once. Constraints (4) define that at most $|K|$ vehicles depart from the depot. Equations (5) are flow conservation constraints, which specify that each arrival forces a departure. The inequalities (6) and (7) are adaptations of the well know *MTZ* subtour elimination constraints for the *TSP* [8]. The capacity of a vehicle must not be exceeded at any time (8). In case customer i is a delivery customer, these constraints specify that the total load arriving at i does not exceeds the maximum capacity of the vehicle by the inequality $y_i + d_i + z_i \leq Q$. Similarly, if i is a pickup customer, the inequality $y_i + z_i + p_i \leq Q$ specifies that the load of the vehicle, when leaving customer i, must be within the capacity limit. Finally (9) are non-negativity and binary restrictions.

Since this formulation is based on the model of Süral-Bookbinder[5] we also adapted the five improvements presented in their paper to the MVRPDSP. We are not going into the details of them, for that please refer to [5].

IV. Hybrid constructive heuristic

The proposed hybrid constructive heuristic is a cluster-first heuristic. Thus it can be separated into two distinct phases: clustering and routing. In order to generate good quality solutions we propose solving the clustering phase by using an adapted Capacitated p-Median mathematical model and the routing phase by using the solution of a TSP and a knapsack problem formulations as base to route each cluster heuristically. It is important to emphasize that, as in the formulation, in case a customer has both demands, it is split into two different customers, one with each demand and the cost to travel between the two is set as zero. Figure 2 shows a framework of the heuristic. Each component is detailed in the following.



Cluster Phase
solved with a capacitated facility location formulation
depot
Routing Phase
for each cluster
Route the delivery customer using a TSP formulation
Heuristically inserts the pickup customers in feasible and profitable positions
Combine the routes to get the solution
Delivery Customer
Pickup Customer
Vehicle 1
Vehicle 2

Figure 2. Example of how our heuristic procedure works

A. *Clustering*

Among the advantages of using an exact approach to solve this phase are the fact that the solution will probably be much better than an heuristic approach, depending on the formulation used, and that any solution generated will be feasible, simplifying the process, since extra effort to generate a feasible solution or to repair it is not necessary. Of course that the downside of such approach can be the runtime needed to optimally solve the model for bigger instances. When the runtime proves to be an issue a feasible solution is acceptable and a time limit for the model can be assigned.

We use the following Capacitated p-Median formulation in order to define the clusters:

$$Min \sum_{i \in V_d} \sum_{j \in V_d} c_{ij} x_{ij} \tag{10}$$

Subject to:

$$\sum_{i \in V_d} x_{ij} = 1 \qquad \forall j \in V_d \tag{11}$$

$$\sum_{j \in V_d} d_j x_{ij} \leq Q y_i \qquad \forall i \in V_d \tag{12}$$

$$\sum_{i \in V_d} y_i = |K| \tag{13}$$

$$x_{ij}, y_i = \{0, 1\} \qquad \forall i, j \in V_d \tag{14}$$

Let $x_{ij} = 1$ if customer j is served by a facility opened at customer i and 0 otherwise. Also consider $y_i = 1$ if a facility was opened at customer i and 0 otherwise. Constraints (11) specify that every customer must be served by one facility. Of course the capacity of the facilities can not be exceed (12), which will guarantee that there is a way to route the customers, within a cluster without overweighting the vehicle. Constraints (13) specify that the number of clusters in the solution will be exactly the same amount of vehicles in the instance. Note that this formulation only considers delivery customers, since their demand is mandatory, thus we must guarantee that they will be served. This also speed up the runtime need to solve the model. In addition, as our test instances contains only customers with both demands and considering that they will be split into two different customers we simply assign pickup customers to the cluster where their respective delivery customers are.

The concept of facility in such model is only used as a base to generate the clusters minimizing the distance between the customers that are part of each cluster. Furthermore it is not guaranteed that the optimal solution for this formulation will group the customers in the same way they appear in the optimal solution of the MVRPDSP, although it usually generates good cluster formations.

B. *Routing*

Once having formed the clusters we must route their customers in order to generate a solution for the MVRPDSP. The routing procedure is based on the one presented in [3], which yielded very good results. In order to take advantage of this heuristic in our problem, we consider each cluster as an instance of the SVRPDSP, which is obviously valid.

For each cluster we apply the routing procedure, which begins by solving a TSP formulation for the delivery customers associated to the cluster being routed and the depot. The basic idea is to route these customers in the best way possible minimizing the transportation cost. Note that, since we considered the vehicle capacity when clustering, there is always a TSP feasible route within the cluster.

Having an initial feasible route, we must take advantage of the pickup customers in the given cluster so as to reduce the total cost of the route. Therefore, to that end, we use a Knapsack Problem formulation to decide which customers have the best cost-benefit. In order to insert the pickup customers of the optimal solution of the Knapsack model, we use an iterative procedure, in which at each iteration all possible positions to insert are evaluated and the best and feasible ones stored in a *Restricted Candidate List* (RCL) of size $rclSize$. Then a randomly chosen candidate of the RCL list is inserted into the current route. This procedure stops when either there are no customers left to insert or there are not a single feasible position to insert the remainder customers.

Applying the described routing procedure to each cluster will generate a set of routes, which is a solution for the MVRPDSP.

V. Computational results

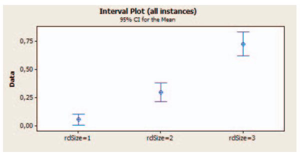
As the present work is the first to address the MVRPDSP, there are no instances available to test our approaches. Therefore we decided to adapted the most used set of instances of the SVRPDSP, which was generated by [6]. This set contains a total of 68 instances, ranging from 15 to 110 customers with both demands, which means that in practice the smallest number of customers is 30. Let $D = \sum_{i \in V_d} d_i$. In order to suit these instances to our problem we assigned the capacity of each vehicle as $\lfloor \alpha D \rfloor$, where $\alpha = \{0.2, 0.4, 0.6\}$. In addition the maximum number of vehicles for each instances is necessary. Since, in a real world context, the number of vehicles is often a critical factor and it is much more expensive to acquire a new vehicle than to perform the deliveries and pickups with fewer vehicles but with a higher cost, we decided to use as few vehicles as possible. Therefore for each instance we calculate the lower bound of the number of vehicles needed to perform all delivery demands. This lower bound is the number of bins in an optimal solution of the well known Bin Packing Problem considering the delivery customers as the item set and the capacity of each bin as the vehicle capacity.

We selected the instances of size 16, 26, 36, 45, 51 and 72 to transform into instances of the MVRPDSP, generating a total of 72 instances.

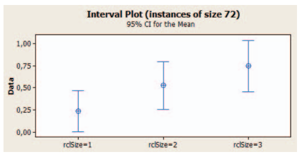
A. Parameter calibration

The constructive algorithm has a parameter $rclSize$, which defines the size of the RCL used in its routing phase. In order to choose the appropriate value for this parameter we collected data from 30 runs of the constructive with $rclSize = \{1, 2, 3\}$ for the 72 instances. This set of values was chosen since [2] performed a similar analysis for their constructive algorithm, which is fairly close to our routing phase, concluding that a great decline in solution quality occurred for values greater than 2 for the RCL size.

Consider $best_i$ as the best solution value found in 30 runs of the constructive with $rclSize = i$ for a given instance and $best = min(best_1, best_2, best_3)$ and $worst = max(best_1, best_2, best_3)$. We normalized the best values using the formula $norm_i = (best_i - best)/(worst - best + 1e - 9)$ resulting in values in the interval 0:1. Having these values normalized we can perform an ANOVA (Analysis of Variance) to statistically analyse which parameter value is the best. The software *Minitab 16* was used for that end. The result of this analysis can be seen in Figure 3(a). Clearly the $rclSize = 1$ is the best value with 95% of confidence. Since for the SVRPDSP the $rclSize = 2$ proved to be a better choice, we decided to perform an ANOVA only considering our biggest instances, with size 72. The results can be seen in Figure 3(b) and shows that the difference between the $rclSize = 1$ and $rclSize = 2$ is much smaller. This behavior could be due to the fact that in bigger instances the solutions tend to have more customers per route, which greatly increases the number of possibilities to insert the pickup customers in the routing phase. Therefore, we foresee that for instances with a greater number of customers the value 2 may be equally a good choice or even a better one. Nevertheless, since the constructive is not time consuming we decided to use both, $rclSize = 1$ and $rclSize = 2$, then choose the best solution generated.



(a) ANOVA for all instances



(b) ANOVA for instances of size 72

Figure 3. Graphs for the Analysis of Variance

B. Results analysis

Initial tests with the model showed that it consumes a lot of memory and, consequently, its execution stops prematurely in several cases, delivering only a feasible solution or none for the most difficult scenarios. For many instances this time was less than an hour, therefore, a time limit of one hour was assigned for each run.

Furthermore, we decided to perform three different trials. At first, we tested the performance of the model, adapting to multiple vehicles all the improvements proposed by [5]. Then, the performance of the constructive algorithm was checked with a single run for $rclSize = 1$ and a total of 30 runs with $rclSize = 2$, selecting the best solution found. Finally we combined both approaches, using the constructive algorithm to generate an initial solution for the model. Note that this solution is the best among all the solutions found in the 31 runs of the constructive. The results of our computational tests are shown in Table I. The instances are named `NNN_B_TYPE_C_ALFA` where `NNN_B_TYPE` is the original name of the instance [6], `NNN` being the number of customers plus the depot, and `ALFA` refers to the α used to set the capacity of each vehicle in our multi-vehicle problem. The results of each method are presented, and also the percentage of improvement that the combined approach achieved over each method alone. Optimal solutions are marked with '*'.

Notice that the constructive algorithm alone outmatched the model in the majority of the cases, even finding the optimal solution for the first 4 instances. Although the model was able to find these optimals, the least time spent in these four runs was more than 10 minutes, while the constructive found the same solution in a matter of milliseconds.

The combination of both approaches showed improvements for most cases, although not leading to new proved optimal solutions. This could be due to a weak lower bound generated by the model and its high consume of memory, which does not allow it to run for a longer time. Comparing the combination *M+C* with the constructive alone the average improvement is of 15% with a maximum of 615%. Compared to the model alone, the combined approach improved all the solutions, except of course the ones already optimal, with an average improvement over 300%. For 5 instances (for example `036_B_one_C_0.6` and `026_B_half_C_0.4`) the model alone found a feasible solution, but the combined approach found a profitable solution, with negative cost, which means that the pickup demands could more efficiently be included in the routes.

All the results were obtained in a Intel(R) Core(TM) i7 3.07GHz machine with 6Gb RAM, running the operating system Ubuntu 12.04. All codes were written in C++ and all formulations were solved using the software *IBM ILOG CPLEX 12.2* under the academic license.

VI. Conclusions

We presented the MVRPDSP along with a formulation and a hybrid constructive heuristic for such problem. This constructive is a cluster-first heuristic and uses exact information to cluster the customers and to route. As to our knowledge this is the first work to address the MVRPDSP, then we proposed a set of 72 instances based on the ones proposed by [6] SVRPDSP.

Table I
RESULTS OF THE PROPOSED APPROACHES

Instance	Model(M)	Construc-tive(C)	Model+Constructive(M+C)		
				%improv over	
				M	C
016_B_p_two_C_0.6	149.94*	149.94*	149.94*	-	-
016_B_half_C_0.6	32.19*	32.19*	32.19*	-	-
016_B_one_C_0.6	-174.27*	-174.27*	-174.27*	-	-
016_B_two_C_0.6	-587.20*	-587.20*	-587.20*	-	-
026_B_p_two_C_0.6	337.42	188.06	184.57	45	2
026_B_half_C_0.6	-98.77	-73.54	-77.09	22	5
026_B_one_C_0.6	-534.74	-509.51	-516.84	3	1
026_B_two_C_0.6	-1406.23	-1381.48	-1406.71	-	2
036_B_p_two_C_0.6	294.39	152.78	152.78	48	-
036_B_half_C_0.6	-70.64	-126.79	-142.78	102	13
036_B_one_C_0.6	161.24	-637.97	-644.34	500	1
036_B_two_C_0.6	-1744.23	-1660.26	-1741.76	-	5
045_B_p_two_C_0.6	907.94	228.89	194.36	79	15
045_B_half_C_0.6	-172.85	-461.41	-493.10	185	7
045_B_one_C_0.6	273.68	-1618.77	-1650.97	703	2
045_B_two_C_0.6	-1137.49	-3933.57	-3965.77	249	1
051_B_p_two_C_0.6	493.30	126.68	126.68	74	-
051_B_half_C_0.6	-34.46	-358.30	-365.16	960	2
051_B_one_C_0.6	-414.62	-1168.98	-1171.95	183	-
051_B_two_C_0.6	-2668.11	-2790.27	-2797.13	5	-
072_B_p_two_C_0.6	518.00	-2.17	-8.00	102	269
072_B_half_C_0.6	—	-329.44	-379.09	—	15
072_B_one_C_0.6	—	-949.13	-949.13	—	-
072_B_two_C_0.6	-525.22	-2180.54	-2231.77	325	2
016_B_p_two_C_0.4	197.81	190.18	187.61	5	1
016_B_half_C_0.4	146.13	80.69	78.12	47	3
016_B_one_C_0.4	-45.87	-112.00	-128.10	179	14
016_B_two_C_0.4	-481.85	-497.40	-542.79	13	9
026_B_p_two_C_0.4	664.31	202.04	195.04	71	3
026_B_half_C_0.4	567.57	-49.09	-56.09	110	14
026_B_one_C_0.4	8.44	-467.62	-472.85	5702	1
026_B_two_C_0.4	-714.08	-1304.71	-1344.82	88	3
036_B_p_two_C_0.4	—	177.92	177.92	—	-
036_B_half_C_0.4	—	-90.97	-95.45	—	5
036_B_one_C_0.4	—	-606.80	-641.00	—	6
036_B_two_C_0.4	—	-1629.09	-1698.42	—	4
045_B_p_two_C_0.4	—	311.15	250.06	—	20
045_B_half_C_0.4	—	-355.05	-433.35	—	22
045_B_one_C_0.4	—	-1507.57	-1616.63	—	7
045_B_two_C_0.4	—	-3822.37	-3958.64	—	4
051_B_p_two_C_0.4	—	162.16	162.16	—	-
051_B_half_C_0.4	—	-296.04	-296.04	—	-
051_B_one_C_0.4	—	-1090.51	-1125.69	—	3
051_B_two_C_0.4	—	-2679.38	-2679.38	—	-
072_B_p_two_C_0.4	—	-16.63	-18.14	—	9
072_B_half_C_0.4	—	-381.47	-381.47	—	-
072_B_one_C_0.4	—	-998.94	-998.94	—	-
072_B_two_C_0.4	—	-2233.91	-2233.91	—	-
016_B_p_two_C_0.2	—	286.74	282.41	—	2
016_B_half_C_0.2	—	177.25	172.92	—	2
016_B_one_C_0.2	—	-15.44	-15.44	—	-
016_B_two_C_0.2	-270.80	-400.85	-405.18	50	1
026_B_p_two_C_0.2	—	347.80	347.80	—	-
026_B_half_C_0.2	—	128.56	104.44	—	19
026_B_one_C_0.2	—	-219.84	-269.91	—	23
026_B_two_C_0.2	—	-987.17	-1098.13	—	11
036_B_p_two_C_0.2	—	284.77	277.89	—	2
036_B_half_C_0.2	—	3.41	-17.55	—	615
036_B_one_C_0.2	—	-507.78	-517.06	—	2
036_B_two_C_0.2	—	-1530.07	-1569.42	—	3
045_B_p_two_C_0.2	—	789.54	783.33	—	1
045_B_half_C_0.2	—	161.38	161.38	—	-
045_B_one_C_0.2	—	-915.25	-915.25	—	-
045_B_two_C_0.2	—	-3068.56	-3068.56	—	-
051_B_p_two_C_0.2	—	306.03	306.03	—	-
051_B_half_C_0.2	—	-139.63	-139.63	—	-
051_B_one_C_0.2	—	-901.67	-901.67	—	-
051_B_two_C_0.2	—	-2425.68	-2425.68	—	-
072_B_p_two_C_0.2	—	241.61	241.61	—	-
072_B_half_C_0.2	—	-64.98	-64.98	—	-
072_B_one_C_0.2	—	-588.88	-588.88	—	-
072_B_two_C_0.2	—	-1815.34	-1815.34	—	-

At first the model was tested, then the constructive and finally a third approach combining the constructive to generate an initial solution for the model was proposed and tested. The results show that the model by itself is not able to even find feasible solutions for several instances, while the constructive for the majority of the cases find better or equal solutions than the model in a matter of milliseconds while the model takes much longer. The combined approach proved to improve in average 15% the solution found by the constructive, while in the best case the improvement reached 615%. This results proves the success of our constructive heuristic and the combination of the heuristic and the model.

Future works include the use of metaheuristics, and a local search to improve the solution, possibly exchanging customers from different routes, and even allowing the delivery and pickup services of the same customer to be served by different routes.

ACKNOWLEDGMENT

This project was supported by CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior.

REFERENCES

[1] J. Privé, J. Renaud, F. Boctor and G. Laporte: Solving a vehicle-routing problem arising in soft-drink distribution. Journal of the Operational Research Society 57, 1045–1052 (2006)

[2] B.P. Bruck, A.G. Santos and J.E.C. Arroyo: Hybrid metaheuristic for the single vehicle routing problem with deliveries and selective pickups. In: 2012 IEEE Congress on Evolutionary Computation. IEEE Press, Brisbane, Australia, 910–917 (2012)

[3] B.P. Bruck, A.G. Santos and J.E.C. Arroyo: An Evolutionary Algorithm and a Variable Neighborhood Descent Algorithm for the Single Vehicle Problem with Deliveries and Selective Pickups. In: XLIV Simpósio Brasileiro de Pesquisa Operacional, Rio de Janeiro, Brazil (2012).

[4] G. Laporte and I. Gribkovskaia: One-to-Many-to-One Single Vehicle Pickup and Delivery Problems. In: Golden, Raghavan, Wasil (eds.) The Vehicle Routing Problems: Latest Advances and New Challenges, pp. 350-377. Springer (2008)

[5] H. Süral, J.H. Bookbinder: The single-vehicle routing problem with unrestricted backhauls. Networks 41, 127–136 (2003)

[6] I. Gribkovskaia, G. Laporte and A. Shyshou: The single vehicle routing problem with deliveries and selective pickups. Computers & Operations Research 35, 2908–2924 (2008)

[7] I. M. Coelho: Contribuições para o problema de roteamento de veículos de rota única com entrega obrigatórias e coletas seletivas. Master thesis, Programa de Pós-Graduação em Computação, Universidade Federal Fluminense, 2011

[8] C.E. Miller, A.W. Tucker, and R.A. Zemlin: Integer programming formulations and travelling salesman problems. Journal of ACM, 326-329 (1960)

[9] D. Vigo: VRPLIB: A Vehicle Routing Problem LIBrary. [Online]. Available: http://www.or.deis.unibo.it/research_pages/ORinstances /VRP LIB/VRPLIB.html