New Lower Bounds for the Vehicle Routing Problem with Simultaneous Pickup and Delivery

Anand Subramanian[1], Eduardo Uchoa[2], and Luiz Satoru Ochi[1]

[1] Universidade Federal Fluminense, Instituto de Computação, Rua Passo da Pátria 156 - Bloco E - 3º andar, Niterói-RJ 24210-240, Brazil
[2] Universidade Federal Fluminense, Departamento de Engenharia de Produção, Rua Passo da Pátria 156 - Bloco E - 4º andar, Niterói-RJ 24210-240, Brazil

Abstract. This work deals with the Vehicle Routing Problem with Simultaneous Pickup and Delivery. We propose undirected and directed two-commodity flow formulations, which are based on the one developed by Baldacci, Hadjiconstantinou and Mingozzi for the Capacitated Vehicle Routing Problem. These new formulations are theoretically compared with the one-commodity flow formulation proposed by Dell'Amico, Righini and Salani. The three formulations were tested within a branch-and-cut scheme and their practical performance was measured in well-known benchmark problems. The undirected two-commodity flow formulation obtained consistently better results. Several optimal solutions to open problems with up to 100 customers and new improved lower bounds for instances with up to 200 customers were found.

1 Introduction

The Vehicle Routing Problem with Simultaneous Pickup and Delivery (VRPSPD) is a variant of the Capacitated Vehicle Routing Problem (CVRP), in which clients require both pickup and delivery services. This problem was first proposed two decades ago by Min [1]. The VRPSPD problem is clearly $\mathcal{NP}$-hard since it can be reduced to the CVRP when all the pickup demands are equal to zero. Practical applications arise especially in the Reverse Logistics context. Companies are increasingly faced with the task of managing the reverse flow of finished goods or raw-materials. Thus, one should consider not only the Distribution Logistics, but also the management of the reverse flow.

The VRPSPD can be defined as follows. Let $G = (V, E)$ be a complete graph with a set of vertices $V = \{0, ..., n\}$, where the vertex 0 represents the depot and the remaining ones the customers. Each edge $\{i, j\} \in E$ has a non-negative cost c_{ij} and each client $i \in V' = V - \{0\}$ has non-negative demands d_i for delivery and p_i for pickup. Let $C = \{1, ..., m\}$ be a set of homogeneous vehicles with capacity Q. The VRPSPD consists in constructing a set up to m routes in such a way that: (i) every route starts and ends at the depot; (ii) all the pickup and delivery demands are accomplished; (iii) the vehicle's capacity is not exceeded in any part of a route; (iv) a customer is visited by only a single vehicle; (v) the sum of costs is minimized.

Although heuristic strategies are by far the most employed to solve the VRPSPD, some exact algorithms were also explored in the literature. A branch-and-price algorithm was developed by Dell'Amico et al. [2], in which two different strategies were used to solve the subpricing problem: (i) exact dynamic programming and (ii) state space relaxation. The authors managed to find optimal solutions for instances with up to 40 clients. Angelelli and Manisini [3] also developed a branch-and-price approach based on the set covering formulation, but for the VRPSPD with time-windows constraints. The subproblem was formulated as a shortest-path with resource constraints but without the elementary condition and it was solved by applying a permanent labeling algorithm. The authors were able find optimal solutions for instances with up to 20 clients. Three-index formulations for the VRPSPD were proposed by Dethloff [4] and Montané and Galvão [5], however only the last authors had tested it in practice. They ran their formulation in CPLEX 9.0 within a time limit of 2 hours and had reported the lower bounds produced for benchmark instances with 50-400 customers.

In this work we propose an undirected and a directed two-commodity flow formulations for the VRPSPD. These formulations extend the one developed by Baldacci et al. [6] for the CVRP. They were compared with the one-commodity flow formulation presented by Dell'Amico et al. [2]. In addition, the three formulations were implemented within a branch-and-cut algorithm, including cuts from the CVRPSEP library [7], and they were tested in well-known benchmark problems with up to 200 customers.

The remainder of this paper is organized as follows. Section 2 describes the one-commodity flow formulation [2]. In Section 3 we present the undirected and the directed two-commodity flow formulations for the VRPSPD and we compare these formulations with the one developed in [2]. Section 4 contains the experimental results obtained by means of a branch-and-cut algorithm. Section 5 presents the concluding remarks.

2 One-commodity flow formulation

Reasonably simple and effective formulations for the CVRP can be defined only over the natural edge variables (arc variables in the asymmetric case), see [8]. Similar formulations are not available for the VRPSPD. This difference between these two problems can be explained as follows. In the CVRP, the feasibility of a route can be determined by only checking whether the sum of its client demands does not exceed the vehicle's capacities. In contrast, the feasibility of a VRPSPD route depends crucially on the sequence of visitation of the clients.

The following directed one-commodity flow formulation for the VRPSPD was proposed by Dell'Amico et al. [2]. Define A as the set of arcs consisting of a pair of opposite arcs (i, j) and (j, i) for each edge $\{i, j\} \in E$ and let D_{ij} and P_{ij} be the flow variables which indicate, respectively, the delivery and pickup loads carried along the arc $(i, j) \in A$. Let x_{ij} be 1 if the arc $(i, j) \in A$ is in the solution and 0 otherwise. The formulation F1C is described next.

$$\min \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \tag{1}$$

subject to

$$\sum_{j \in V} x_{ij} = 1 \qquad \forall i \in V' \tag{2}$$

$$\sum_{j \in V} x_{ji} = 1 \qquad \forall i \in V' \tag{3}$$

$$\sum_{j \in V'} x_{0j} \leq m \tag{4}$$

$$\sum_{j \in V} D_{ji} - \sum_{j \in V} D_{ij} = d_i \qquad \forall i \in V' \tag{5}$$

$$\sum_{j \in V} P_{ij} - \sum_{j \in V} P_{ji} = p_i \qquad \forall i \in V' \tag{6}$$

$$D_{ij} + P_{ij} \leq Q x_{ij} \qquad \forall (i,j) \in A \tag{7}$$

$$D_{ij} \geq 0 \qquad \forall (i,j) \in A \tag{8}$$

$$P_{ij} \geq 0 \qquad \forall (i,j) \in A \tag{9}$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \tag{10}$$

The objective function (1) minimizes the sum of the travel costs. Constraints (2)-(3) impose that each client should be visited exactly once. Constraints (4) refer to the number of vehicles available. Constraints (5)-(7) are the flow conservation equalities. Constraints (8)-(10) define the domain of the decision variables.

Dell'Amico et al. [2] basically extended the one-commodity flow formulation proposed by Gavish and Graves [9] for the CVRP by adding constraints (6) and (9), and the term P_{ij} in (7). Gouveia [10] showed that it is possible to obtain stronger inequalities for D_{ij} by using the tighter bounds (11) instead of (8) in the Gavish and Graves formulation. Accordingly, we can apply the same idea to develop stronger inequalities for P_{ij} by replacing (9) with (12) and for $D_{ij} + P_{ij}$ by replacing (7) with (13).

$$d_j x_{ij} \leq D_{ij} \leq (Q - d_i) x_{ij} \qquad \forall (i,j) \in A \tag{11}$$

$$p_i x_{ij} \leq P_{ij} \leq (Q - p_j) x_{ij} \qquad \forall (i,j) \in A \tag{12}$$

$$D_{ij} + P_{ij} \leq (Q - \max\{0, p_j - d_j, d_i - p_i\}) x_{ij} \qquad \forall (i,j) \in A \tag{13}$$

It should be noticed that a lower bound for (13) is implicit in (11) and (12), i.e, $D_{ij} + P_{ij} \geq d_j x_{ij} + p_i x_{ij}$. Another valid inequality for F1C, given by (14), is due to the fact that each edge not adjacent to the depot is traversed at most once.

$$x_{ij} + x_{ji} \leq 1 \qquad \forall i, j, i < j, \in V' \tag{14}$$

3 Two-commodity flow formulations

In this section we present both an undirected and a directed two-commodity flow formulations for the VRPSPD which are based on the one proposed by Baldacci et al. [6] for the CVRP.

3.1 Undirected two-commodity flow formulation

For the sake of convenience let vertex $n+1$ be a copy of the depot, $\bar{V} = V \cup \{n+1\}$ and $\bar{E}$ be the complete set of edges $\bar{E}$, excepting $\{0, n+1\}$. Let x'_{ij} be 1 if the edge $\{i, j\} \in \bar{E}$ is in the solution and 0 otherwise. Let the variables D'_{ij}, P'_{ij} and SPD_{ij} denote, respectively, the delivery, pickup and simultaneous pickup and delivery flows when a vehicle goes from $i \in \bar{V}$ to $j \in \bar{V}$ and let the same variables denote, respectively, the associated residual capacities when a vehicle goes from $j \in \bar{V}$ to $i \in \bar{V}$, in such a way that $D'_{ij} + D'_{ji} = Qx'_{ij}$, $P'_{ij} + P'_{ji} = Qx'_{ij}$ and $SPD_{ij} + SPD_{ji} = Qx'_{ij}$. Also, an integer variable v, which denotes the number of vehicles utilized, is included with an upper bound m.

The undirected formulation F2C-U is as follows.

$$\min \sum_{\{i,j\} \in \bar{E}} c_{ij} x'_{ij} \tag{15}$$

subject to

$$\sum_{i \in \bar{V}, i<k} x'_{ik} + \sum_{j \in \bar{V}, j>k} x'_{kj} = 2 \qquad \forall k \in V' \tag{16}$$

$$\sum_{j \in \bar{V}} (D'_{ji} - D'_{ij}) = 2d_i \qquad \forall i \in V' \tag{17}$$

$$\sum_{j \in V'} D'_{0j} = \sum_{i \in V'} d_i \tag{18}$$

$$\sum_{j \in V'} D'_{j0} = vQ - \sum_{i \in V'} d_i \tag{19}$$

$$\sum_{j \in \bar{V}} (P'_{ij} - P'_{ji}) = 2p_i \qquad \forall i \in V' \tag{20}$$

$$\sum_{j \in V'} P'_{jn+1} = \sum_{i \in V'} p_i \tag{21}$$

$$\sum_{j \in V'} P'_{n+1j} = vQ - \sum_{i \in V'} p_i \tag{22}$$

$$\sum_{j \in \bar{V}} (SPD_{ji} - SPD_{ij}) = 2(d_i - p_i) \qquad \forall i \in V' \tag{23}$$

$$SPD_{0j} = D'_{0j} \qquad \forall j \in V' \tag{24}$$

$$SPD_{j0} = D'_{j0} \qquad \forall j \in V' \tag{25}$$

$$SPD_{jn+1} = P'_{jn+1} \qquad \forall j \in V' \tag{26}$$

$$SPD_{n+1j} = D'_{n+1j} \qquad \forall j \in V' \tag{27}$$

$$D'_{ij} + D'_{ji} = Qx'_{ij} \qquad \forall \{i,j\} \in \bar{E} \tag{28}$$

$$P'_{ij} + P'_{ji} = Qx'_{ij} \qquad \forall \{i,j\} \in \bar{E} \tag{29}$$

$$SPD_{ij} + SPD_{ji} = Qx'_{ij} \qquad \forall \{i,j\} \in \bar{E} \tag{30}$$

$$D'_{jn+1} = P'_{0j} = 0 \qquad \forall j \in V' \tag{31}$$

$$\sum_{j \in V'} D'_{n+1j} = \sum_{j \in V'} P'_{j0} = vQ \tag{32}$$

$$\sum_{j \in V'} x'_{0j} = \sum_{j \in V'} x'_{jn+1} = v \tag{33}$$

$$0 \leq v \leq m \tag{34}$$

$$D'_{ij} \geq 0, D'_{ji} \geq 0 \qquad \forall \{i,j\} \in \bar{E} \tag{35}$$

$$P'_{ij} \geq 0, P'_{ji} \geq 0 \qquad \forall \{i,j\} \in \bar{E} \tag{36}$$

$$SPD_{ij} \geq 0, SPD_{ji} \geq 0 \qquad \forall \{i,j\} \in \bar{E} \tag{37}$$

$$x'_{ij} \in \{0,1\} \qquad \forall (i,j) \in \bar{E} \tag{38}$$

The objective function (15) minimizes the sum of the travel costs. Constraints (16) are the degree equations. Constraints (17) ensure that the delivery demands are satisfied. Constraints (18) state that the sum of the vehicle loads leaving the vertex 0 must be equal to the sum of the demand of all costumers. Constraints (19) enforce that the sum of the vehicle loads arriving at the vertex 0 must be equal to the sum of the residual capacity of all vehicles. Constraints (20)-(22) are related to the pickup flow and their meaning are, respectively, analogous to (17)-(19). Constraints (23) guarantee that the pickup and delivery demands are simultaneously satisfied. Constraints (24)-(27) are self-explanatory. Constraints (28)-(30) state, respectively, that the sum of the delivery, pickup and combined loads arriving and leaving each customer must be equal to the vehicle capacity. Constraints (31)-(32) are self-explanatory. Constraint (33) is related to the number of vehicles. Constraints (34)-(38) define the domain of the decision variables.

The formulation F2C-U was obtained by simply adding constraints (20)-(27), (29)-(34) and (36)-(37) to the formulation presented in [6]. As in F1C, stronger inequalities can be developed by tightening the bounds of the flow variables, i.e, replacing (35)-(36) with (39)-(40) and (37) with (41).

$$D'_{ij} \geq d_j x'_{ij} \qquad \forall (i,j) \in \bar{E} \tag{39}$$

$$P'_{ij} \geq p_i x'_{ij} \qquad \forall (i,j) \in \bar{E} \tag{40}$$

$$SPD_{ij} \geq \max\{0, d_j - p_j, p_i - d_i\} x'_{ij} \qquad \forall (i,j) \in \bar{E} \tag{41}$$

Although the lower bounds of the flow variables are not explicit in (39)-(41) it can be easily verified that they become inherent to the formulation when these upper bound inequalities are combined with (28)-(30), resulting in $D'_{ij} \leq (Q - d_i)x'_{ij}$, $P'_{ij} \leq (Q - d_j)x'_{ij}$ and $SPD_{ij} \leq (Q - \max\{0, d_i - p_i, p_j - d_j\})x'_{ij}$.

3.2 Directed two-commodity flow formulation

Let $\bar{A}$ be the set of arcs (i,j), $\forall i,j \in \bar{V}$ and $\bar{x}_{ij}$ be 1 if the arc $(i,j) \in \bar{A}$ is in the solution and 0 otherwise. A directed version of the two-commodity flow formulation (F2C-D) is as follows.

$$\min \sum_{i\in\bar{V}} \sum_{j\in\bar{V}} c_{ij}\bar{x}_{ij} \tag{42}$$

subject to

$$\sum_{j\in\bar{V}} \bar{x}_{ij} = 1 \qquad \forall i \in V' \tag{43}$$

$$\sum_{j\in\bar{V}} \bar{x}_{ji} = 1 \qquad \forall i \in V' \tag{44}$$

$$\bar{x}_{j0} = \bar{x}_{n+1j} = 0 \qquad \forall j \in V' \tag{45}$$

$$D'_{ij} + D'_{ji} = Q(\bar{x}_{ij} + \bar{x}_{ji}) \qquad \forall (i,j), i<j, \in A \tag{46}$$

$$P'_{ij} + P'_{ji} = Q(\bar{x}_{ij} + \bar{x}_{ji}) \qquad \forall (i,j), i<j, i \neq 0 \in \bar{A} \tag{47}$$

$$SPD_{ij} + SPD_{ji} = Q(\bar{x}_{ij} + \bar{x}_{ji}) \qquad \forall i,j, i<j, \in V' \tag{48}$$

$$\sum_{j\in V'} \bar{x}_{0j} = \sum_{j\in V'} \bar{x}_{jn+1} = v \tag{49}$$

$$\bar{x}_{ij} \in \{0,1\} \qquad \forall (i,j) \in \bar{A} \tag{50}$$

(17)-(32) and (34)-(37)

Constraints (43)-(44) are the degree equations. Constraint (45) is self-explanatory. Constraints (46)-(48) are the capacity equalities. Constraints (49)-(50) have already been defined.

The stronger flow inequalities defined for the F2C-U also hold for the F2C-D as can be observed in (51)-(53). Also, the arc inequalities (14) used in F1C can be directly converted to F2C-D as shown in (54).

$$D'_{ij} \geq d_j(\bar{x}_{ij} + \bar{x}_{ji}) \qquad \forall (i,j) \in \bar{A} \tag{51}$$

$$P'_{ij} \geq p_i(\bar{x}_{ij} + \bar{x}_{ji}) \qquad \forall (i,j) \in \bar{A} \tag{52}$$

$$SPD_{ij} \geq (\max\{0, d_j - p_j, p_i - d_i\})(\bar{x}_{ij} + \bar{x}_{ji}) \qquad \forall (i,j) \in \bar{A} \tag{53}$$

$$\bar{x}_{ij} + \bar{x}_{ji} \leq 1 \qquad \forall i,j, i<j, \in V' \tag{54}$$

Proposition 1. *The linear relaxation of F1C with (11)-(14) is strictly stronger than the one obtained by F2C-D with (51)-(54), which in turn is strictly stronger than the the linear relaxation of F2C-U with (39)-(41).*

4 Computational Experiments with a branch-and-cut algorithm

We evaluated the practical performance of the formulations presented in this work when used in a branch-and-cut (BC) algorithm. Traditional CVRP inequalities were used, namely the rounded capacity, multistar and comb inequalities. The cuts were separated using the CVRPSEP package [7]. The reader is referred to [11] for details concerning the separation routines. At first, we try to separate the cuts using the delivery demands. When no valid inequalities are found we then use the pickup demands. All of the three kinds of cuts are generated at the root node, but just the rounded capacity cuts are used throughout the tree up to the 5th level. Preliminary tests have shown that the overhead of separating comb and multistar inequalities outside the root node was not worthwhile. For each separation routine of the CVRPSEP package we have established a limit of 50 violated cuts per iteration.

The BC procedures were implemented using the CPLEX 11.2 callable library and executed in an Intel Core 2 Quad with 2.4 GHz and 4 GB of RAM running under Linux 64 bits (kernel 2.6.27-16). Only a single thread was used in our experiments. Each BC is respectively associated with the formulations F1C, F2C-U and F2C-D and they were tested on the set of instances proposed by Dethloff [4], Salhi and Nagy [12] and Montané and Galvão [5]. The first group contains 40 instances with 50 customers, the second contains 14 instances with 50-199 customers, while the third contains 12 instances with 100-200 customers. The number of vehicles is not explicitly specified in these 66 instances. The barrier algorithm was used to solve the initial linear relaxation of the last two group of instances. A time limit of 2 hours of execution was imposed for the BC algorithms. In some very particular cases, the CPLEX have slightly exceeded this time limit, namely on few instances involving more than 100 customers. The values of the best known solutions found in the literature were given as initial primal bound for the BC, namely those reported in [13].

In the tables presented hereafter, ***#v*** represents the number of vehicles in the best known solution, **LP** is the linear relaxation, **Root LB** indicates the root lower bound, after CVRPSEP cuts are added, **Root Time** is the CPU time in seconds spent at the root node, **Tree size** corresponds to the the number of nodes opened, **Total time** is the total CPU time in seconds of the BC procedure, **Prev. LB** is the lower bound obtained in [5], **New LB** is the best lower bound determined among the three flow formulations, **F-LB** is the lower bound found by the respective formulation, **UB** is the upper bound reported in [13], and **Gap** corresponds to the gap between the LB and the UB. Proven optimal solutions are highlighted in boldface. If the F-LB is the one associated with the New LB (F-LB = New LB), then its value is underlined only if New LB is not an optimal solution.

Tables 1, 2 and 3 contain, respectively, the results obtained by F1C, F2C-U and F2C-D on the set of instances of Dethloff. It can be seen that the three formulations were able to prove the optimality of almost all instances of 4 vehicles. F2C-D appears to be the most effective under this aspect, being capable of proving the optimality of 18 instances. The performance of the three formulations on the instances of 9 vehicles were inferior in terms of optimality proof, but their LBs are significantly better then the previous values reported in [5]. F2C-U seems to be the most effective in terms of LBs, with an average gap of 0.94%, against 1.34% and 1.19% of F1C and F2C-D, respectively.

Table 1. Results obtained by the F1C on Dethloff's instances

Instance/ Customers	#v	LP	Root LB	Root Time (s)	Tree size	Total Time (s)	Prev. LB	New LB	F-LB	UB	Gap (%)
SCA3-0/50	4	551.14	613.38	41	73473	7200	583.77	625.89	622.73	635.62	2.03
SCA3-1/50	4	645.95	682.40	107	1712	1230	655.63	697.84	**697.84**	**697.84**	0.00
SCA3-2/50	4	592.56	658.35	19	1	19	627.12	659.34	659.34	659.34	0.00
SCA3-3/50	4	586.30	667.37	70	1083	415	633.56	680.04	**680.04**	**680.04**	0.00
SCA3-4/50	4	627.29	672.92	80	8718	1599	642.89	690.50	**690.50**	**690.50**	0.00
SCA3-5/50	4	604.31	646.14	83	15560	1901	603.06	659.90	**659.90**	**659.90**	0.00
SCA3-6/50	4	587.97	624.92	47	37655	7200	607.53	644.37	639.97	651.09	1.71
SCA3-7/50	4	584.69	654.30	82	26	103	616.40	659.17	**659.17**	**659.17**	0.00
SCA3-8/50	4	638.75	688.77	94	72785	7200	668.04	719.48	703.12	719.48	2.27
SCA3-9/50	4	597.02	668.09	82	1674	417	619.03	681.00	**681.00**	**681.00**	0.00
SCA8-0/50	9	849.35	922.36	96	8854	7200	877.55	936.89	933.12	961.50	2.95
SCA8-1/50	9	937.71	998.04	75	9948	7200	954.29	1020.28	1015.05	1049.65	3.30
SCA8-2/50	9	931.93	1008.83	78	10334	7200	950.74	1024.24	1019.99	1039.64	1.89
SCA8-3/50	9	874.31	954.55	74	11375	7200	905.29	983.34	970.88	983.34	1.27
SCA8-4/50	9	958.58	1022.44	72	12054	7200	972.62	1041.65	1036.45	1065.49	2.73
SCA8-5/50	9	923.50	996.01	79	9207	7200	940.60	1015.78	1011.57	1027.08	1.51
SCA8-6/50	9	870.58	933.57	133	6219	7200	885.34	959.91	944.53	971.82	2.81
SCA8-7/50	9	937.30	1013.86	65	12533	7200	955.86	1031.56	1029.97	1051.28	2.03
SCA8-8/50	9	962.50	1023.86	102	8510	7200	986.52	1048.93	1036.90	1071.18	3.20
SCA8-9/50	9	953.36	1012.73	89	9031	7200	978.90	1034.28	1031.51	1060.50	2.73
CON3-0/50	4	577.74	606.00	91	40753	4836	592.38	616.52	616.46	616.52	0.01
CON3-1/50	4	506.41	543.71	73	52033	6498	532.55	554.47	**554.47**	**554.47**	0.00
CON3-2/50	4	468.40	503.14	61	13874	7200	491.04	518.00	514.11	518.00	0.75
CON3-3/50	4	541.46	581.45	55	20044	1941	557.99	591.19	**591.19**	**591.19**	0.00
CON3-4/50	4	537.90	577.61	63	78398	7200	558.26	588.79	588.47	588.79	0.06
CON3-5/50	4	511.88	553.87	107	32652	5975	531.33	563.70	**563.70**	**563.70**	0.00
CON3-6/50	4	468.90	486.59	128	14248	7200	475.33	499.05	493.01	499.05	1.21
CON3-7/50	4	533.86	562.10	38	53629	5522	550.73	576.48	**576.48**	**576.48**	0.00
CON3-8/50	4	477.81	513.90	87	15317	1923	492.69	523.05	**523.05**	**523.05**	0.00
CON3-9/50	4	528.34	564.87	63	15461	5602	547.31	578.25	**578.25**	**578.25**	0.00
CON8-0/50	9	774.69	829.80	47	16498	7200	795.45	845.19	842.62	857.17	1.70
CON8-1/50	9	680.24	719.03	80	7552	7200	693.22	734.71	732.44	740.85	1.14
CON8-2/50	9	636.18	682.76	128	9856	7200	650.81	695.70	693.07	712.89	2.78
CON8-3/50	10	732.55	784.93	71	7536	7200	754.41	797.57	796.31	811.07	1.82
CON8-4/50	9	710.36	749.83	122	6374	7200	729.09	767.63	759.11	772.25	1.70
CON8-5/50	9	696.85	728.10	78	7901	7200	709.76	741.51	736.79	754.88	2.40
CON8-6/50	9	611.16	647.04	61	10400	7200	631.41	662.14	662.14	678.92	2.47
CON8-7/50	9	729.28	787.89	64	11861	7200	762.03	810.08	800.22	811.96	1.44
CON8-8/50	9	689.23	741.02	74	10324	7200	705.08	757.45	753.42	767.53	1.84
CON8-9/50	9	716.21	770.66	101	5435	7200	729.10	786.40	778.65	809.00	3.75
										Avg. Gap (%)	1.34

In order to check if the values of the UB of the instances SCA3-0, SCA3-6, SCA8-6, CON8-1, CON8-4 and CON8-7 are optimal we ran F2C-U with a time limit of 48 hours. The formulation was successful to prove the optimality of each of these instances within up to 36 hours of execution.

As for the Salhi and Nagy and Montané and Galvão instances, we will present only the results obtained by F2C-U, not only because it produced the best results

Table 2. Results obtained by the F2C-U on Dethloff's instances

Instance/ Customers	#v	LP	Root LB	Root Time (s)	Tree size	Total Time (s)	Prev. LB	New LB	F-LB	UB	Gap (%)
SCA3-0/50	4	550.85	613.36	28	211456	7200	583.77	625.89	625.00	635.62	1.67
SCA3-1/50	4	645.59	682.33	42	1467	142	655.63	697.84	**697.84**	**697.84**	0.00
SCA3-2/50	4	592.44	658.89	12	1	12	627.12	659.34	**659.34**	**659.34**	0.00
SCA3-3/50	4	586.02	667.37	25	847	81	633.56	680.04	**680.04**	**680.04**	0.00
SCA3-4/50	4	626.93	673.28	50	2866	252	642.89	690.50	**690.50**	**690.50**	0.00
SCA3-5/50	4	603.95	646.29	38	19724	731	603.06	659.90	**659.90**	**659.90**	0.00
SCA3-6/50	4	587.85	624.89	27	176561	7200	607.53	644.37	644.37	651.09	1.03
SCA3-7/50	4	584.59	653.76	34	30	43	616.40	659.17	**659.17**	**659.17**	0.00
SCA3-8/50	4	638.41	693.71	42	108017	4899	668.04	719.48	**719.48**	**719.48**	0.00
SCA3-9/50	4	596.79	668.09	36	1452	96	619.03	681.00	**681.00**	**681.00**	0.00
SCA8-0/50	9	847.39	922.58	79	17695	7200	877.55	936.89	936.89	961.50	2.56
SCA8-1/50	9	933.44	997.07	56	18086	7200	954.29	1020.28	1020.28	1049.65	2.80
SCA8-2/50	9	931.34	1008.27	75	12789	7200	950.74	1024.24	1024.24	1039.64	1.48
SCA8-3/50	9	872.37	953.67	49	18487	7200	905.29	983.34	975.87	983.34	0.76
SCA8-4/50	9	955.74	1021.35	44	25853	7200	972.62	1041.65	1041.65	1065.49	2.24
SCA8-5/50	9	922.25	995.93	58	19464	7200	940.60	1015.78	1013.87	1027.08	1.29
SCA8-6/50	9	868.00	933.76	74	10467	7200	885.34	959.91	959.91	971.82	1.23
SCA8-7/50	9	935.55	1015.11	69	15193	7200	955.86	1031.56	1031.56	1051.28	1.88
SCA8-8/50	9	960.17	1023.60	87	8262	7200	986.52	1048.93	1048.93	1071.18	2.08
SCA8-9/50	9	952.34	1014.89	64	16262	7200	978.90	1034.28	1034.28	1060.50	2.47
CON3-0/50	4	577.52	606.82	46	3048	247	592.38	616.52	**616.52**	**616.52**	0.00
CON3-1/50	4	506.23	545.53	54	16039	823	532.55	554.47	**554.47**	**554.47**	0.00
CON3-2/50	4	468.22	504.44	59	22107	7200	491.04	518.00	516.23	518.00	0.34
CON3-3/50	4	541.40	582.83	35	6608	330	557.99	591.19	**591.19**	**591.19**	0.00
CON3-4/50	4	537.73	577.57	42	50663	3198	558.26	588.79	**588.79**	**588.79**	0.00
CON3-5/50	4	511.59	554.35	65	10191	729	531.33	563.70	**563.70**	**563.70**	0.00
CON3-6/50	4	468.75	486.61	100	48466	5230	475.33	499.05	**499.05**	**499.05**	0.00
CON3-7/50	4	533.73	561.87	37	9822	1141	550.73	576.48	**576.48**	**576.48**	0.00
CON3-8/50	4	477.45	514.13	71	5541	450	492.69	523.05	**523.05**	**523.05**	0.00
CON3-9/50	4	527.94	564.78	53	6372	790	547.31	578.25	**578.25**	**578.25**	0.00
CON8-0/50	9	773.46	827.14	74	13038	7200	795.45	845.19	845.19	857.17	1.40
CON8-1/50	9	678.95	719.09	67	13302	7200	693.22	734.71	734.71	740.85	0.83
CON8-2/50	9	635.23	682.37	127	9409	7200	650.81	695.70	695.70	712.89	2.41
CON8-3/50	10	731.55	785.00	71	18680	7200	754.41	797.57	797.57	811.07	1.66
CON8-4/50	9	708.64	751.32	60	15700	7200	729.09	767.63	767.63	772.25	0.60
CON8-5/50	9	696.08	727.26	66	9765	7200	709.76	741.51	741.51	754.88	1.77
CON8-6/50	9	610.20	646.78	94	11947	7200	631.41	662.14	661.36	678.92	2.59
CON8-7/50	9	726.55	788.64	74	5520	7200	762.03	810.08	810.08	811.96	0.23
CON8-8/50	9	688.25	741.76	81	13325	7200	705.08	757.45	757.45	767.53	1.31
CON8-9/50	9	713.85	770.85	109	12833	7200	729.10	786.40	786.40	809.00	2.79
									Avg. Gap (%)		0.94

on average, but also due to lack of space. From Table 4 it can be observed that optimality of the instances CMT1X and CMT1Y has been proven. In addition, to our knowledge these are the first LBs presented for these set of instances. Montané and Galvão [5] had reported LBs for the case where the demands were rounded to the nearest integer. From Table 5 it can be noticed that optimality of the instances r201, c201 and rc201 was proven. The main characteristic of these three instances is the fact of having relatively very few vehicles.

Table 6 shows a summary of the results obtained by the three formulations in all set of instances. In this table, **G1** is the average gap between the linear relaxation and the UB, **G2** is the average gap with respect to the root LB, including the CVRPSEP cuts, and **G3** is average gap for the LB, possibly after branching, found within the time limit established. Those results can be explained as

Table 3. Results obtained by the F2C-D on Dethloff's instances

Instance/ Customers	#v	LP	Root LB	Root Time (s)	Tree size	Total Time (s)	Prev. LB	New LB	F-LB	UB	Gap (%)
SCA3-0/50	4	550.93	613.35	46	78840	7200	583.77	625.89	625.89	635.62	1.53
SCA3-1/50	4	645.60	682.32	57	1296	197	655.63	697.84	**697.84**	**697.84**	0.00
SCA3-2/50	4	592.47	658.73	11	1	11	627.12	659.34	**659.34**	**659.34**	0.00
SCA3-3/50	4	586.02	667.37	25	398	68	633.56	680.04	**680.04**	**680.04**	0.00
SCA3-4/50	4	626.93	672.91	39	6185	395	642.89	690.50	**690.50**	**690.50**	0.00
SCA3-5/50	4	603.96	647.68	61	9304	396	603.06	659.90	**659.90**	**659.90**	0.00
SCA3-6/50	4	587.85	624.89	32	128038	7200	607.53	644.37	643.89	651.09	1.11
SCA3-7/50	4	584.59	654.17	34	78	42	616.40	659.17	**659.17**	**659.17**	0.00
SCA3-8/50	4	638.41	694.96	39	50016	2132	668.04	719.48	**719.48**	**719.48**	0.00
SCA3-9/50	4	596.79	668.09	32	3155	166	619.03	681.00	**681.00**	**681.00**	0.00
SCA8-0/50	9	847.73	922.10	78	5194	7200	877.55	936.89	934.35	961.50	2.82
SCA8-1/50	9	933.46	999.17	76	7976	7200	954.29	1020.28	1012.75	1049.65	3.52
SCA8-2/50	9	931.42	1009.17	114	3355	7200	950.74	1024.24	1019.72	1039.64	1.92
SCA8-3/50	9	872.44	954.83	100	14625	5435	905.29	983.34	**983.34**	**983.34**	0.00
SCA8-4/50	9	955.96	1022.17	90	5833	7200	972.62	1041.65	1035.21	1065.49	2.84
SCA8-5/50	9	922.31	996.17	158	16121	7200	940.60	1015.78	1015.78	1027.08	1.10
SCA8-6/50	9	868.05	933.70	171	2092	7200	885.34	959.91	941.94	971.82	3.07
SCA8-7/50	9	935.81	1013.96	105	4632	7200	955.86	1031.56	1027.82	1051.28	2.23
SCA8-8/50	9	960.27	1023.40	159	1857	7200	986.52	1048.93	1033.51	1071.18	3.52
SCA8-9/50	9	952.40	1012.90	119	4124	7200	978.90	1034.28	1027.24	1060.50	3.14
CON3-0/50	4	577.52	605.93	30	19230	1503	592.38	616.52	**616.52**	**616.52**	0.00
CON3-1/50	4	506.24	543.58	32	9539	720	532.55	554.47	**554.47**	**554.47**	0.00
CON3-2/50	4	468.22	503.22	37	14378	1186	491.04	518.00	**518.00**	**518.00**	0.00
CON3-3/50	4	541.40	581.45	30	12099	584	557.99	591.19	**591.19**	**591.19**	0.00
CON3-4/50	4	537.73	577.57	22	59287	3064	558.26	588.79	**588.79**	**588.79**	0.00
CON3-5/50	4	511.60	553.94	36	14671	950	531.33	563.70	**563.70**	**563.70**	0.00
CON3-6/50	4	468.75	486.47	71	44447	7200	475.33	499.05	498.05	499.05	0.20
CON3-7/50	4	533.75	562.79	28	42432	3689	550.73	576.48	**576.48**	**576.48**	0.00
CON3-8/50	4	477.45	514.07	44	7460	480	492.69	523.05	**523.05**	**523.05**	0.00
CON3-9/50	4	527.95	564.76	48	4623	903	547.31	578.25	**578.25**	**578.25**	0.00
CON8-0/50	9	773.51	826.68	92	3116	7200	795.45	845.19	840.86	857.17	1.90
CON8-1/50	9	679.00	719.35	89	4981	7200	693.22	734.71	729.55	740.85	1.53
CON8-2/50	9	635.25	682.25	128	3037	7200	650.81	695.70	692.85	712.89	2.81
CON8-3/50	10	731.55	784.65	94	7403	7200	754.41	797.57	796.30	811.07	1.82
CON8-4/50	9	708.64	751.49	58	9224	7200	729.09	767.63	766.15	772.25	0.79
CON8-5/50	9	696.08	726.63	60	4283	7200	709.76	741.51	739.39	754.88	2.05
CON8-6/50	9	610.20	646.58	72	4706	7200	631.41	662.14	656.80	678.92	3.26
CON8-7/50	9	726.56	787.95	56	3934	7200	762.03	810.08	802.14	811.96	1.21
CON8-8/50	9	688.33	740.72	71	11026	7200	705.08	757.45	753.63	767.53	1.81
CON8-9/50	9	713.93	770.12	115	7032	7200	729.10	786.40	781.45	809.00	3.41
									Avg. Gap (%)		1.19

follows. The linear relaxation of is F1C is indeed a little better than the linear relaxations of F2C-D and F2C-U. However, after the cuts, there is no significant difference in the LB quality. This can be clearly seen in the column **G2** under Dethloff instances. For those smaller instances, the cut separation in the root node could always be completed within the time limit. In those cases, the small gap differences (2.96%, 2.94% and 2.92%) are not significant and can be attributed to the heuristic nature of the routines in the CVRPSEP library. The consistent advantage of formulation F2C-U shown in columns **G3** is explained by the fact that CPLEX has a significantly better performance when reoptimizing its LPs. This means that more cuts can be separated and more nodes can be explored within the same time limit.

Table 4. Results obtained by the F2C-U on Salhi and Nagy's instances

Instance/ Customers	$\#v$	LP	Root LB	Root Time (s)	Tree size	Total Time (s)	New LB	F-LB	UB	Gap (%)
CMT1X/50	3	449.00	459.98	102	2282	300	466.77	**466.77**	**466.77**	0.00
CMT1Y/50	3	449.00	460.02	70	3205	213	466.77	**466.77**	**466.77**	0.00
CMT2X/75	6	632.11	652.85	346	2073	7200	655.98	655.21	684.21	4.24
CMT2Y/75	6	632.11	653.13	449	2610	7200	655.73	655.41	684.21	4.21
CMT3X/100	5	682.18	701.10	504	13820	7200	705.39	704.35	721.27	2.35
CMT3Y/100	5	682.18	701.12	612	19865	7200	705.28	705.28	721.27	2.22
CMT12X/100	5	564.08	628.59	813	991	7201	629.39	629.39	662.22	4.96
CMT12Y/100	5	564.08	628.58	923	118	7201	629.18	629.09	662.22	5.00
CMT11X/120	4	687.42	775.51	4835	42	7201	776.35	776.35	833.92	6.90
CMT11Y/120	4	687.42	775.40	6138	22	7200	775.74	775.74	833.92	6.98
CMT4X/150	7	796.48	817.11	7288	1	7292	817.11	817.11	852.46	4.15
CMT4Y/150	7	796.48	816.99	5747	1	7201	817.06	816.99	852.46	4.16
CMT5X/200	10	933.21	954.87	6939	1	7201	954.87	954.87	1029.25	7.23
CMT5Y/200	10	933.21	953.56	6600	1	7202	953.56	953.56	1029.25	7.35
								Avg. Gap (%)		4.27

Table 5. Results obtained by the F2C-U on Montané and Galvão's instances

Instance/ Customers	$\#v$	LP	Root LB	Root Time (s)	Tree size	Total Time (s)	Prev. LB	New LB	F-LB	UB	Gap (%)
r101/100	12	939.19	972.88	2910	122	7201	934.97	973.17	973.10	1009.95	3.65
r201/100	3	643.07	664.80	292	21	307	643.65	666.20	**666.20**	**666.20**	0.00
c101/100	16	1070.40	1195.53	1396	302	7201	1066.19	1197.48	1195.89	1220.18	1.99
c201/100	5	598.47	657.97	197	17	241	596.85	662.07	**662.07**	**662.07**	0.00
rc101/100	10	944.21	1028.15	2940	138	7201	937.41	1029.38	1029.38	1059.32	2.83
rc201/100	3	600.24	671.84	134	4	134	602.70	672.92	**672.92**	**672.92**	0.00
r1_2_1/200	23	3013.16	3084.97	6971	1	7200	2951.12	3084.97	3084.97	3360.02	8.19
r2_2_1/200	5	1549.60	1618.76	7869	1	7874	1501.82	1631.38	1618.76	1665.58	2.81
c1_2_1/200	28	3325.20	3475.03	7041	1	7202	3299.07	3475.03	3475.03	3629.89	4.27
c2_2_1/200	9	1560.22	1647.83	7370	1	7374	1542.96	1647.83	1647.83	1726.59	4.56
rc1_2_1/200	23	3015.44	3093.30	7064	1	7201	2939.98	3093.30	3093.30	3306.00	6.43
rc2_2_1/200	5	1438.91	1551.07	7301	1	7308	1396.95	1551.07	1551.07	1560.00	0.57
									Avg. Gap (%)		2.94

Table 6. Summary of the results obtained by the three formulations

Formulation	Dethloff			Salhi and Nagy			Montané and Galvão		
	G1 (%)	G2 (%)	G3 (%)	G1 (%)	G2 (%)	G3 (%)	G1 (%)	G2 (%)	G3 (%)
F1C	9.74	2.96	1.34	9.21	4.85	4.57	8.75	3.66	3.57
F2C-D	9.85	2.94	1.19	9.30	4.66	4.32	8.82	3.57	3.48
F2C-U	9.85	2.92	0.94	9.30	4.62	4.27	8.82	3.04	2.94

5 Concluding Remarks

This work dealt with Mixed Integer Programming formulations for the the Vehicle Routing Problem with Simultaneous Pickup and Delivery. An undirected and a directed two-commodity flow formulations were proposed. They were tested within a branch-and-cut scheme and their results were compared with the one-commodity flow formulation of Dell'Amico et al. [2]. The optimal solutions of 30 open problems were proved, and new lower bounds were obtained for instances with up to 200 customers. In addition, although we have shown that the one-commodity flow formulation produces a stronger linear relaxation, the two-commodity flow formulations have found, on average, better lower bounds after 2 hours of execution time.

References

1. Min, H.: The multiple vehicle routing problem with simultaneous delivery and pick-up points. Transportation Research **23**(5) (1989) 377–386
2. Dell'Amico, M., Righini, G., Salani, M.: A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection. Transportation Science **40**(2) (2006) 235–247
3. Angelelli, E., Mansini, R.: A branch-and-price algorithm for a simultaneous pick-up and delivery problem. In: Quantitative Approaches to Distribution Logistics and Supply Chain Management. Springer, Berlin-Heidelberg (2002) 249–267
4. Dethloff, J.: Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up. OR Spektrum **23** (2001) 79–96
5. Montané, F.A.T., Galvão, R.D.: A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. European Journal of Operational Research **33**(3) (2006) 595–619
6. Baldacci, R., Hadjiconstantinou, E., Mingozzi, A.: An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. Operations Research **52**(5) (2004) 723–738
7. Lysgaard, J.: A package of separation routines for the capacited vehicle routing problem. Technical report, available at www.asdb.dk/∼lys (2003)
8. Toth, P., Vigo, D.: Models, relaxations and exact approaches for the capacitated vehicle routing problem. Discrete Applied Mathematics **123** (2002) 487–512
9. Gavish, B., Graves, S.: The traveling salesman problem and related problems. Working Paper (1979)
10. Gouveia, L.: A result on projection for the vehicle routing problem. European Journal of Operational Research **85** (1995) 610–624
11. Lysgaard, J., Letchford, A.N., Eglese, R.W.: A new branch-and-cut algorithm for the capacitated vehicle routing problem. Mathematical Programming **100** (2003) 423–445
12. Salhi, S., Nagy, G.: A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. Journal of the Operational Research Society **50** (1999) 1034–1042
13. Subramanian, A., Drummond, L.M.A., Bentes, C., Ochi, L.S., Farias, R.: A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. Computers & Operations Research (2010 (to appear))

Appendix

In this appendix we present the proof of Proposition 1.

Proof. First we shall prove that given the solution vector (x^*, D^*, P^*) with cost z^* of the linear programming relaxation of the one-commodity flow formulation, it is possible to build a feasible solution of the linear program of F2C-D (in terms of $(\bar{x}, D', P', SPD, v)$) with the same cost.

The values of the variables of F2C-D can be directly obtained by means of (55)-(68). For the sake of simplicity let $P_{jn+1} = P_{j0}$ and $P_{n+1j} = P_{0j}$, $\forall j \in V'$.

$$\bar{x}_{ij} = x_{ij} \qquad \forall i,j \in V' \qquad (55)$$

$$\bar{x}_{0j} = x_{0j} \qquad \forall j \in V' \qquad (56)$$

$$\bar{x}_{jn+1} = x_{j0} \qquad \forall j \in V' \qquad (57)$$

$$D'_{ij} = D_{ij} + (Q\bar{x}_{ji} - D_{ji}) \qquad \forall (i,j), i < j, \in A \qquad (58)$$

$$D'_{ji} = D_{ji} + (Q\bar{x}_{ij} - D_{ij}) \qquad \forall (i,j), i < j, \in A \qquad (59)$$

$$P'_{ij} = P_{ij} + (Q\bar{x}_{ji} - P_{ji}) \qquad \forall (i,j), i < j, i \neq 0 \in \bar{A} \qquad (60)$$

$$P'_{ji} = P_{ji} + (Q\bar{x}_{ij} - P_{ij}) \qquad \forall (i,j), i < j, i \neq 0 \in \bar{A} \qquad (61)$$

$$SPD_{ij} = D_{ij} + P_{ij} + (Q\bar{x}_{ji} - D_{ji} - P_{ji}) \qquad \forall i,j, i < j, \in V' \qquad (62)$$

$$SPD_{ji} = D_{ji} + P_{ji} + (Q\bar{x}_{ij} - D_{ij} - P_{ij}) \qquad \forall i,j, i < j, \in V' \qquad (63)$$

$$D'_{jn+1} = P'_{0j} = \bar{x}_{j0} = \bar{x}_{n+1j} = 0 \qquad \forall j \in V' \qquad (64)$$

$$D'_{n+1j} = Q\bar{x}_{n+1j}, P'_{j0} = Q\bar{x}_{j0} \qquad \forall j \in V' \qquad (65)$$

$$SPD_{0j} = D'_{0j}, SPD_{j0} = D'_{j0} \qquad \forall j \in V' \qquad (66)$$

$$SPD_{jn+1} = P'_{jn+1}, SPD_{n+1j} = P'_{n+1j} \qquad \forall j \in V' \qquad (67)$$

$$v = \sum_{j \in V'} \bar{x}_{0j} \qquad (68)$$

Note that constraints (46)-(48) are automatically satisfied since they can be easily obtained from (58)-(63). Constraints (51) are satisfied since, according to (11), $D_{ij} \geq d_j\bar{x}_{ij}$ and $Q\bar{x}_{ji} - D_{ji} \geq d_j\bar{x}_{ji}$, which implies in $D'_{ij} \geq d_j(\bar{x}_{ij} + \bar{x}_{ji})$. The same idea can be employed, using (12), to show that constraints (52) are also satisfied.

To verify if constraints (53) are not violated we need to prove the following statement: $D_{ij} + P_{ij} + (Q\bar{x}_{ij} - D_{ji} - P_{ji}) \geq (\max\{0, d_j - p_j, p_i - d_i\})(\bar{x}_{ij} + \bar{x}_{ji})$. Using the fact that $D_{ij} + P_{ij} \geq d_j\bar{x}_{ij} + p_i\bar{x}_{ij}$ (see (11)-(12)) and after some algebraic manipulation we obtain: $d_j\bar{x}_{ij} + p_i\bar{x}_{ij} + (Q - \max\{0, d_j - p_j, p_i - di\})\bar{x}_{ji} \geq D_{ji} + P_{ji} + (\max\{0, d_j - p_j, p_i - d_i\})\bar{x}_{ij}$. From (13) we observe that $(Q - \max\{0, d_j - p_j, p_i - di\})\bar{x}_{ji} \geq D_{ji} + P_{ji}$ and it is clear that $d_j\bar{x}_{ij} + p_i\bar{x}_{ij} \geq (\max\{0, d_j - p_j, p_i - d_i\})\bar{x}_{ij}$, which proves that (53) is satisfied.

Thus we conclude that the vector $(\bar{x}, D', P', SPD, v)$ is indeed a feasible solution of the linear program of F2C-D.

On the other hand, given the solution vector $(\bar{x}^*, D'^*, P'^*, SPD^*, v^*)$ with cost $\bar{z}^*$ of the linear programming relaxation of F2C-D it is not always possible to build a feasible solution in terms of (x, D, P) with the same cost. Tables 1 and 3, presented in Section 4, show instances where the value of the linear relaxation obtained by the F1C is strictly greater than the one found by F2C-D.

Formulations F2C-D and F2C-U are very similar. It can be seen that equalities (43)-(44) are disaggregated versions of equalities (16). Therefore, the linear relaxation of F2C-D is at least as strong as the linear relaxation of F2C-U. Table 3 show instances where F2C-D obtained slightly stronger lower bounds.