Branch-and-cut with lazy separation for the Vehicle Routing Problem with Simultaneous Pickup and Delivery

Anand Subramanian[*,a,b], Eduardo Uchoa[c], Artur Alves Pessoa[c], Luiz Satoru Ochi[b]

[a] *Universidade Federal da Paraíba, Departamento de Engenharia de Produção, Brazil*
[b] *Universidade Federal Fluminense, Instituto de Computação, Brazil*
[c] *Universidade Federal Fluminense, Departamento de Engenharia de Produção, Brazil*

Abstract

We propose a branch-and-cut algorithm for the VRPSPD where the constraints that assure that the capacities are not exceeded in the middle of a route are applied in a lazy fashion. The algorithm was tested in 87 instances with 50-200 customers, finding improved lower bounds and several new optimal solutions.

Key words: Vehicle Routing, Simultaneous Pickup and Delivery, Branch-and-cut

1. Introduction

The Vehicle Routing Problem with Simultaneous Pickup and Delivery (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 customer $i \in V' = V - \{0\}$ has non-negative demands d_i for delivery and p_i for pickup. Consider a fleet of identical vehicles with capacity Q. The VRPSPD consists in constructing a set of routes in such a way that: (i) every route starts and ends at the depot; (ii) all the pickup and delivery demands are attended; (iii) the vehicle's capacity is not exceeded in any part of a route; (iv) each customer is visited exactly once; (v) the sum of costs is minimized. The VRPSPD is $\mathcal{NP}$-hard since it includes the Capacitated Vehicle Routing Problem (CVRP) as a special case when all the pickup (or delivery) demands are equal to zero. A particular case of the VRPSPD, known as the Vehicle Routing Problem with Mixed Pickup and Delivery (VRPMPD), arises when customers either have a pickup or a delivery demand, that is, if $d_i > 0$, then $p_i = 0$ and vice-versa. In general, any solution approach developed for the VRPSPD can also be applied to the VRPMPD.

Real-life applications of the VRPSPD can be found especially within the Reverse Logistics context. Companies are increasingly faced with the task of managing the reverse flow of finished goods or raw-materials. Therefore, one should take into account not only the Distribution Logistics, but also the management of the reverse flow. Both the Distribution Logistic and Reverse Logistic should act together with an aim to guarantee the synchronization between the pickup and delivery

*Corresponding author: Anand Subramanian. Universidade Federal da Paraíba, Departamento de Engenharia de Produção, Centro de Tecnologia, Campus I - Bloco G, Cidade Universitária, João Pessoa-PB, 58051-970, Brazil.
E-mail addresses: anand@ic.uff.br, anandsubraman@gmail.com

operations, as well as their impact on the company's supply chain, resulting in the customer's satisfaction and minimization of the operational efforts.

The VRPSPD received a lot of attention from the literature in the past decade. This problem has been mostly tackled by heuristic algorithms and, to our knowledge, the most effective in terms of solution quality are the parallel Iterated Local Search proposed in [1] and the local search based metaheuristic developed in [2]. Some exact algorithms can also be found in the VRPSPD literature. A branch-and-price algorithm was suggested in [3], where the authors managed to find optimal solutions for instances with up to 40 customers. A branch-and-cut (BC) approach over extended flow formulations that makes use of the CVRPSEP library [4] was recently proposed in [5, 6]; it was capable of proving the optimality of instances with up to 100 customers.

In this work we propose a BC algorithm, that also includes cuts from the CVRPSEP library, over a formulation with only the edge variables. The developed solution approach was tested in well-known VRPSPD/VRPMPD instances with up to 200 customers and it was capable of improving most of the previously known lower bounds.

2. Mathematical Formulation

Let x_{ij} be an integer variable counting the number of times that a vertex $\{i, j\} \in E$ appears in a route, this number can only be 2 for an edge used in a route with a single customer. Given a set $S \subseteq V'$, let $d(S)$ and $p(S)$ be the sum of the delivery and pickup demands, respectively, of all customers in S. Let $e(S) = \lceil d(S)/Q \rceil$ and $q(S) = \lceil p(S)/Q \rceil$. Finally, let v be an integer variable representing the number of vehicles in the solution. Define the following IP:

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

$$\text{s.t.} \sum_{i \in V, i < k} x_{ik} + \sum_{j \in V, j > k} x_{kj} = 2 \qquad \forall k \in V' \tag{2}$$

$$\sum_{j \in V'} x_{0j} = 2v \tag{3}$$

$$\sum_{i \in S} \sum_{j \in \bar{S}, i < j} x_{ij} + \sum_{i \in \bar{S}} \sum_{j \in S, i < j} x_{ij} \geq 2e(S) \qquad \forall S \subseteq V' \tag{4}$$

$$\sum_{i \in S} \sum_{j \in \bar{S}, i < j} x_{ij} + \sum_{i \in \bar{S}} \sum_{j \in S, i < j} x_{ij} \geq 2q(S) \qquad \forall S \subseteq V' \tag{5}$$

$$v \in \mathbb{Z}_+ \qquad \forall \{i, j\} \in E \tag{6}$$

$$x_{ij} \in \{0, 1\} \qquad \forall \{i, j\} \in E, i > 0 \tag{7}$$

$$x_{ij} \in \{0, 1, 2\} \qquad \forall \{0, j\} \in E. \tag{8}$$

This is not a complete formulation. While Constraints (4) and (5) are enough to ensure that all vehicles leave and return to the depot with load at most Q, it is possible that a vehicle capacity may be exceeded in the middle of the route. In fact, previous VRPSPD formulations use auxiliary flows for controlling vehicle load ([3, 5, 6]). Such additional variables and constraints have the drawback of increasing the solution time of the associated LPs. The new formulation proposed in this paper eliminates those unfeasible routes with constraints over the edge variables. Let $\mathcal{R}$

be the set of all routes, represented by their middle edges (non-adjacent to the depot), that are feasible with respect to both pickup and delivery alone, but not with respect to the simultaneous pickup and delivery. The following constraints are added to (1-8) in order to obtain a complete formulation, called F1:

$$\sum_{\{i,j\} \in R} x_{ij} \leq |R| - 1 \qquad \forall R \in \mathcal{R} \tag{9}$$

Constraints similar to (9) are sometimes called "no-good cuts" since they only remove a single infeasible integral point, which is usually weak in a polyhedral sense, because these kind of cuts are seldom violated by fractional solutions. In our case, a constraint associated to an unfeasible route R actually eliminates all integral solutions that contain such route. Nevertheless, they are still weak and only worthy to be applied in a lazy fashion in a BC algorithm, with the purpose of checking the feasibility of the integral solutions found along the tree.

3. Computational Experiments with a branch-and-cut approach

A BC algorithm over F1 was implemented. Besides capacity inequalities (4-5), two other families of valid CVRP inequalities, namely the multistar and comb inequalities, are also separated using the CVRPSEP package [4, 7]. Firstly, we perform the separation considering only the delivery demands (like in Constraints (4)). When no violated inequalities are found we then start separating considering the pickup demands (like in (5)). For each separation routine of the CVRPSEP package we have established a limit of 50 violated cuts per iteration. The multistar and comb inequalities are generated only at the root node. The rounded capacity cuts (4-5) are generated throughout the tree up to the 7th level and whenever an integer solution is found. Moreover, every time a feasible integer CVRP solution is found we check whether it is also feasible for the VRPSPD. If it is not the case we then add inequality (9) for each unfeasible route R.

The BC was 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. Only a single thread was used in our tests.

3.1. VRPSPD

Three set of test-problems are available in the VRPSPD literature, particularly those of Dethloff [8], Salhi and Nagy [9] and Montané and Galvão [10]. The first set contains 40 instances with 50 customers, the second 14 instances with 50-199 customers and the third 18 instances with 100-400 customers, but we only considered those up to 200 customers (12 instances). The number of vehicles is not fixed in none of these set of instances.

In the tables presented hereafter, $\boldsymbol{v}$ is the number of vehicles in the best known solution, **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** is the number of nodes opened, **Total time** is the total CPU time in seconds of the BC procedure, **#Lazy Cuts** denotes the number of lazy cuts (9) added, **Prev. LB** is the lower bound obtained in [5, 6], **Our LB** is the lower bound (LB) determined by our BC, **UB** is the upper bound reported in [1] for the VRPSPD and in [11] for the VRPMPD, and **Gap** corresponds to the gap between the LB and the UB. Proven optimal solutions are highlighted in boldface and new optimal solutions are underlined.

Tables 1-3 present the results obtained by the BC on the set of instances of Dethloff [8], Salhi and Nagy [9] and Montané and Galvão [10], respectively. For the first set of instances, we let the BC run until the optimal solution was found, whereas a time limit of 2 hours of execution was imposed for the remaining sets.

From Table 1, it is possible to observe that, except for the instance CON8-9, the optimality of all instances were proved within up to 5000 seconds. In the instances that require less vehicles the BC spent at most 33 seconds to find the optimal solution. Furthermore, 15 new optimal solutions were proved. From Table 2 it can be seen that, except for the instance CMT5Y, all the LBs were equaled or improved. Also, the optimality of the instances CMT3X and CMT3Y, involving 100 customers, were proved. Lastly, from Table 3, we can verify that the BC equaled or improved the LBs of the instances involving 100 customers, regardless of the number of vehicles. However, for the instances involving 200 customers the results were rather inconsistent. On one hand, the BC found the optimal solutions for the instances requiring few vehicles. On the other hand, the BC obtained poor LBs for the instances requiring a large number of vehicles.

3.2. VRPMPD

A set of 21 VRPMPD instances involving 50-199 customers was proposed by Salhi and Nagy [9]. As in the VRPSPD, the number of vehicles is not specified. Also, Gajpal and Abad [11] did not mention the number of vehicles in their best solutions. A time limit of 2 hours was imposed for the BC (which was sometimes exceeded by CPLEX), except for the instances CMT3H, CMT3T and CMT11Q, where we ran the BC algorithm until the optimal solution was found. From Table 4, we can verify that all LBs were either equaled or improved and one new optimal solution was found (CMT11Q).

4. Concluding Remarks

The success of the proposed BC when compared to previous approaches can be attributed to the use of a formulation where the load of a vehicle in the middle of a route is only controlled by weak constraints, that are only separated in a lazy way over integral solutions, avoiding complicated and larger extended formulations. In practice (at least on the instances from the literature), it seems that there are relatively few routes where the load capacity is respected in both ends but not in the middle, as can be observed by the small number of lazy cuts added throughout the BC (see Tables 1-4).

Acknowledgments

This research was partially supported by the following brazilian research agencies: CNPq, CAPES and FAPERJ. The latter under grants E-26/110.550/2010 and E-26/110.552/2010.

References

[1] A. Subramanian, L. M. A. Drummond, C. Bentes, L. S. Ochi, R. Farias, A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery, Comp. & Oper. Res. 37 (11) (2010) 1899–1911.

[2] E. E. Zachariadis, C. T. Kiranoudis, A local search metaheuristic algorithm for the vehicle routing problem with simultaneous pick-ups and deliveries, Expert Syst. with Appl. 38 (3) (2011) 2717–2726.

[3] M. Dell'Amico, G. Righini, M. Salani, A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection, Transp. Sci. 40 (2) (2006) 235–247.
[4] J. Lysgaard, A package of separation routines for the capacited vehicle routing problem, Tech. rep., available at www.asdb.dk/∼lys (2003).
[5] A. Subramanian, E. Uchoa, L. S. Ochi, New lower bounds for the vehicle routing problem with simultaneous pickup and delivery, in: Proceedings of the 9th International Symposium on Experimental Algorithms (SEA), Lect. Notes in Comp. Sci., 2010, pp. 276–287.
[6] A. Subramanian, E. Uchoa, L. S. Ochi, New lower bounds for the vehicle routing problem with simultaneous pickup and delivery, Tech. Rep. 01/10, Universidade Federal Fluminense, Niterói, Brazil (2010).
[7] J. Lysgaard, A. N. Letchford, R. W. Eglese, A new branch-and-cut algorithm for the capacitated vehicle routing problem, Math. Program. 100 (2004) 423–445.
[8] J. Dethloff, Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up, OR Spektrum 23 (2001) 79–96.
[9] S. Salhi, G. Nagy, A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling, J. Oper. Res. Soc. 50 (1999) 1034–1042.
[10] F. A. T. Montané, R. D. Galvão, A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service, Comp. & Oper. Res. 33 (3) (2006) 595–619.
[11] Y. Gajpal, P. Abad, An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup, Comp. & Oper. Res. 36 (12) (2009) 3215–3223.

Table 1: Results obtained on the instances of Dethloff [8]

Instance/ Customers	v	Root LB	Root Time (s)	Tree size	Total Time (s)	#Lazy Cuts	Prev. LB	Our LB	UB	Gap (%)
SCA3-0/50	4	619.21	5.18	543	16.8	5	635.62	635.62	**635.62**	0.00
SCA3-1/50	4	682.34	1.90	129	6.75	0	697.84	697.84	**697.84**	0.00
SCA3-2/50	4	659.34	0.60	1	0.61	0	659.34	659.34	**659.34**	0.00
SCA3-3/50	4	667.56	1.76	35	2.75	0	680.04	680.04	**680.04**	0.00
SCA3-4/50	4	676.56	6.75	98	10.4	0	690.50	690.50	**690.50**	0.00
SCA3-5/50	4	647.90	1.16	623	5.71	26	659.90	659.91	**659.91**	0.00
SCA3-6/50	4	625.60	1.29	2655	239	0	651.09	651.09	**651.09**	0.00
SCA3-7/50	4	654.97	3.70	5	3.87	0	659.17	659.17	**659.17**	0.00
SCA3-8/50	4	688.21	1.97	5880	710	0	719.48	719.48	**719.48**	0.00
SCA3-9/50	4	671.07	2.04	27	2.88	0	681.00	681.00	**681.00**	0.00
SCA8-0/50	9	926.03	18.8	2836	557	0	936.89	961.50	**961.50**	0.00
SCA8-1/50	9	1002.29	15.5	14292	3288	2	1020.28	1049.65	**1049.65**	0.00
SCA8-2/50	9	1013.37	19.8	4321	553	0	1024.24	1039.64	**1039.64**	0.00
SCA8-3/50	9	956.10	11.9	350	66.5	0	983.34	983.34	**983.34**	0.00
SCA8-4/50	9	1027.24	25.9	9136	931	1	1041.65	1065.49	**1065.49**	0.00
SCA8-5/50	9	998.05	13.7	948	149	3	1015.19	1027.08	**1027.08**	0.00
SCA8-6/50	9	948.26	12.8	750	95.3	12	971.82	971.82	**971.82**	0.00
SCA8-7/50	9	1018.20	16.4	6303	600	4	1031.56	1051.28	**1051.28**	0.00
SCA8-8/50	9	1038.86	6.15	957	100	1	1048.93	1071.18	**1071.18**	0.00
SCA8-9/50	9	1019.25	12.0	9068	1381	1	1034.28	1060.50	**1060.50**	0.00
CON3-0/50	4	611.30	2.58	21	3.39	0	616.52	616.52	**616.52**	0.00
CON3-1/50	4	546.65	8.26	142	13.7	0	554.47	554.47	**554.47**	0.00
CON3-2/50	4	505.06	3.94	293	22.4	6	518.01	518.01	**518.01**	0.00
CON3-3/50	4	584.28	2.05	22	2.62	0	591.19	591.19	**591.19**	0.00
CON3-4/50	4	577.52	1.04	322	10.4	5	588.79	588.79	**588.79**	0.00
CON3-5/50	4	552.91	3.22	350	20.2	2	563.70	563.70	**563.70**	0.00
CON3-6/50	4	486.98	5.22	352	32.5	0	499.05	499.05	**499.05**	0.00
CON3-7/50	4	561.62	0.97	494	29.1	0	576.48	576.48	**576.48**	0.00
CON3-8/50	4	514.84	5.03	97	8.55	3	523.05	523.05	**523.05**	0.00
CON3-9/50	4	564.78	2.51	175	19.4	0	578.25	578.25	**578.25**	0.00
CON8-0/50	9	830.03	41.5	1760	272	0	845.19	857.17	**857.17**	0.00
CON8-1/50	9	722.38	27.7	1179	200	1	740.85	740.85	**740.85**	0.00
CON8-2/50	9	685.66	39.2	14315	4842	1	695.70	712.89	**712.89**	0.00
CON8-3/50	10	787.84	39.3	12377	1599	2	797.57	811.07	**811.07**	0.00
CON8-4/50	9	751.95	21.2	806	159	3	772.25	772.25	**772.25**	0.00
CON8-5/50	9	729.92	15.1	6369	1212	0	741.51	754.88	**754.88**	0.00
CON8-6/50	9	648.64	18.7	10738	2865	0	662.14	678.92	**678.92**	0.00
CON8-7/50	9	793.50	15.5	228	38.4	2	811.96	811.96	**811.96**	0.00
CON8-8/50	9	744.52	26.8	3520	606	0	757.45	767.53	**767.53**	0.00
CON8-9/50	9	773.65	45.0	90738	38137	4	786.40	809.00	**809.00**	0.00

Table 2: Results obtained on the VRPSPD instances of Salhi and Nagy [9]

Instance/ Customers	v	Root LB	Root Time (s)	Tree size	Total Time (s)	#Lazy Cuts	Prev. LB	Our LB	UB	Gap (%)
CMT1X/50	3	460.73	5.56	78	10.3	0	466.77	466.77	**466.77**	0.00
CMT1Y/50	3	460.69	15.5	60	19.3	0	466.77	466.77	**466.77**	0.00
CMT2X/75	6	658.38	100	17718	7200	0	655.98	666.57	684.21	2.65
CMT2Y/75	6	658.12	200	19875	7200	0	655.41	666.69	684.21	2.63
CMT3X/100	5	711.77	79.7	8521	2461	36	705.54	721.27	**721.27**	0.00
CMT3Y/100	5	711.89	140	9313	3226	36	705.62	721.27	**721.27**	0.00
CMT12X/100	5	635.52	50.3	10460	7200	1	629.39	643.76	662.22	2.87
CMT12Y/100	5	635.45	65.1	11621	7200	0	629.18	644.10	662.22	2.81
CMT11X/120	4	793.11	892	5988	7200	0	776.35	799.67	833.92	4.28
CMT11Y/120	4	793.54	1098	4063	7200	0	775.74	799.02	833.92	4.37
CMT4X/150	7	826.74	1950	4012	7200	0	817.11	831.18	852.46	2.56
CMT4Y/150	7	826.34	2568	4380	7200	0	816.99	831.65	852.46	2.50
CMT5X/200	10	971.07	7202	1	7206	0	954.87	971.07	1029.25	5.99
CMT5Y/200	10	937.66	7248	1	7252	0	953.56	937.66	1029.25	9.77

Table 3: Results obtained on the instances of Montané and Galvão [10]

Instance/ Customers	v	Root LB	Root Time (s)	Tree size	Total Time (s)	#Lazy Cuts	Prev. LB	Our LB	UB	Gap (%)
r101/100	12	972.58	1282	5528	7200	0	973.91	978.28	1009.95	3.24
r201/100	3	664.80	42.9	10	43.7	0	666.20	666.20	**666.20**	0.00
c101/100	16	1194.69	722	10480	7200	0	1196.70	1202.59	1220.18	1.46
c201/100	5	657.97	24.6	12	28.3	0	662.07	662.07	**662.07**	0.00
rc101/100	10	1028.06	1067	8208	7200	0	1029.38	1041.06	1059.32	1.75
rc201/100	3	671.84	6.37	3	6.42	0	672.92	672.92	**672.92**	0.00
r1_2_1/200	23	2681.05	7198	1	7202	0	3084.97	2681.05	3360.02	25.32
r2_2_1/200	5	1656.62	4477	974	5551	0	1618.76	1665.58	<u>**1665.58**</u>	0.00
c1_2_1/200	27	2857.45	7196	1	7201	0	3475.03	2857.45	3629.89	27.03
c2_2_1/200	9	1638.99	7197	1	7203	0	1647.83	1638.99	1726.59	5.34
rc1_2_1/200	23	2656.78	7196	1	7201	0	3093.30	2656.78	3306.00	24.44
rc2_2_1/200	5	1551.54	1932	483	2323	0	1551.07	1560.00	<u>**1560.00**</u>	0.00

Table 4: Results obtained on the VRPMPD instances of Salhi and Nagy [9]

Instance/ Customers	v	Root LB	Root Time (s)	Tree size	Total Time (s)	#Lazy Cuts	Prev. LB	Our LB	UB	Gap (%)
CMT1H/50	3	460.10	7.14	103	9.67	10	465.02	465.02	**465.02**	0.00
CMT1Q/50	4	488.15	5.65	5	5.87	0	489.74	489.74	**489.74**	0.00
CMT1T/50	5	512.61	4.98	126	11.7	0	520.06	520.06	**520.06**	0.00
CMT2H/75	-	643.28	101	15003	7200	0	647.84	653.79	662.63	1.33
CMT2Q/75	-	707.66	100	12256	7200	0	711.30	717.36	732.76	2.10
CMT2T/75	-	759.92	151	15585	7200	0	764.99	770.35	782.77	1.59
CMT3H/100	3	691.32	58.0	58219	42606	481	694.92	700.94	**700.94**	0.00
CMT3Q/100	6	744.50	95.5	39	113	0	747.15	747.15	**747.15**	0.00
CMT3T/100	5	784.13	342	92974	164046	0	787.12	798.07	**798.07**	0.00
CMT12H/100	-	623.00	40.3	17791	3018	180	629.37	629.37	**629.37**	0.00
CMT12Q/100	-	721.81	83.9	1577	686	4	729.25	729.25	**729.25**	0.00
CMT12T/100	9	787.27	37.7	3	37.9	0	787.52	787.52	**787.52**	0.00
CMT11H/120	-	801.24	353	4296	7200	0	801.05	806.46	820.35	1.69
CMT11Q/120	6	928.28	653	95065	209428	9	928.74	939.36	**939.36**	0.00
CMT11T/120	-	984.81	359	7232	7200	0	985.03	989.32	998.80	0.95
CMT4H/150	-	799.19	1122	3544	7200	0	798.38	804.10	831.39	3.28
CMT4Q/150	-	892.99	2920	3243	7200	0	890.12	898.28	913.93	1.71
CMT4T/150	-	953.41	4246	1873	7200	0	950.59	956.54	990.39	3.42
CMT5H/199	-	931.02	7209	1	7215	0	922.88	931.02	992.37	6.18
CMT5Q/199	-	1056.74	7206	1	7213	0	1040.25	1056.74	1134.72	6.87
CMT5T/199	-	1145.61	7192	1	7200	0	1139.93	1145.63	1232.08	7.02