VI
Computational Experiments

Over the years the VRPTW has received a lot of attention in the literature. As a consequence of this great interest, various benchmark instance sets have been proposed. One of the most relevant of those benchmarks is the one proposed by Solomon (SOLOMON, 1987).

If the capacity of the vehicles are not considered, this benchmark give us a suitable testbed to experiment the proposed algorithms in this work. For this experiment only one type of vehicle was considered, so no synchronization is needed. We left the minimization of the number of vehicles out of our problem, thus we used the known best values in terms of the used vehicles to set a fixed size fleet in our problem.

We mainly focused in two types of tests. A first round of tests were intended to try out our proposed column generation approach. To this end, we ran tests over all Solomon instances up to 50 customers, solving the Dantzig-Wolfe Master LP.

To run a second round of tests we derived a small group of instances from a subset of the 25 Customer Solomon benchmark. The main objective of the second round of experiments was to compare the difficulty in solving these subset of instances, both with and without synchronization.

All computational experiments of this dissertation were executed on a 3.2Ghz Intel Core i5 computer with 8 GB of RAM. Algorithms where implemented in C++ using MS Visual Studio 2010. All algorithms were executed using the generic MIP solver Gurobi v5.5

This chapter describes the method used to adapt the VRPTW instances to contain synchronization requirements and presents the results of our computational experiments.

VI.1 Results - Round 1

In the first round of experiments, the main intention was to test the flow model and the decomposition approach suggested in this dissertation, for solving instances of VRPTW, which correspond to the special case of the VRPTWEOS with only one type of vehicle. Tests were made over instances up to 50 customers. The results of first round of experiments are shown in tables VI.1 to VI.4 and figures VI.1 to VI.6.

(a) Result Tables

The headers of the result tables of the first round of experiments refer to:

– Table columns under the **Best Known** label, correspond to the optimal solution values to those instances found in the literature.

– Table columns under the **Standard Column Generation** label, correspond to data obtained with the mentioned column generation approach.

– Table columns under the **Gurobi Root Node** correspond to data obtained by solving the root relaxation of the flow original variable formulation with the generic Gurobi Solver.

– The header **Instance** refers to the name of the used Solomon instance.

– The header **V** is the number of vehicles used to solve the instance.

– Header **TD** refers to the total travel distance found in a solution.

– **Gap** header refers to the difference between the optimal travel distance and the travel distance found with the solution approach.

– **Total Routes** make reference to the total number of generated routes (columns) in the column generation approach.

– Table header **Integer** indicates whether an integer (optimal) solution was found while solving the correspondent LP program.

– Header **Time** refers to the total running time in seconds of the solution approach.

Table VI.1:
Results Comparison for Solomons C1 and C2 instances with 25 Customers (No Synchronization)

	Best Known		**Standard Column Generation**					**Gurobi Root Node**			
Instance	**V**	**TD**	**TD**	**Gap**	**Total Routes**	**Integer**	**Time (s)**	**TD**	**Gap**	**Integer**	**Time (s)**
C101	3	191,3	191,8	0,27%	300	Yes	2,3	191,8	0,27%	Yes	2,6
C102	3	190,3	190,7	0,23%	783	Yes	28,9	190,7	0,23%	Yes	36,5
C103	3	190,3	189,2	0,58%	572	No	50,1	188,3	1,06%	No	606,0
C104	3	186,9	186,7	0,09%	671	No	77,1	187,4	0,29%	Yes	622,9
C105	3	191,3	191,8	0,27%	189	Yes	2,9	191,8	0,27%	Yes	6,6
C106	3	191,3	191,8	0,27%	229	Yes	1,8	191,8	0,27%	Yes	3,5
C107	3	191,3	191,8	0,27%	195	Yes	4,2	191,8	0,27%	Yes	10,0
C108	3	191,3	191,8	0,27%	504	Yes	14,0	191,8	0,27%	Yes	38,5
C109	3	191,3	190,9	0,21%	476	No	20,9	190,1	0,63%	No	129,6
C201	2	214,7	215,5	0,39%	1.436	Yes	28,8	215,5	0,39%	Yes	12,3
C202	2	214,7	215,5	0,39%	1.430	Yes	214,7	215,5	0,39%	Yes	119,6
C203	2	214,7	215,5	0,39%	1.646	Yes	477,8	215,5	0,39%	Yes	649,8
C204	2	213,1	215,3	1,04%	1.819	Yes	734,7	214,2	0,53%	No	1590,8
C205	2	214,7	212,9	0,86%	1.070	No	42,0	214,6	0,05%	No	96,7
C206	2	214,7	207,3	3,55%	1.170	No	64,4	212,3	1,12%	No	176,5
C207	2	214,5	211,0	1,67%	1.200	No	102,8	206,8	3,74%	No	454,2
C208	2	214,5	205,7	4,28%	1.056	No	86,3	208,5	2,90%	No	441,5

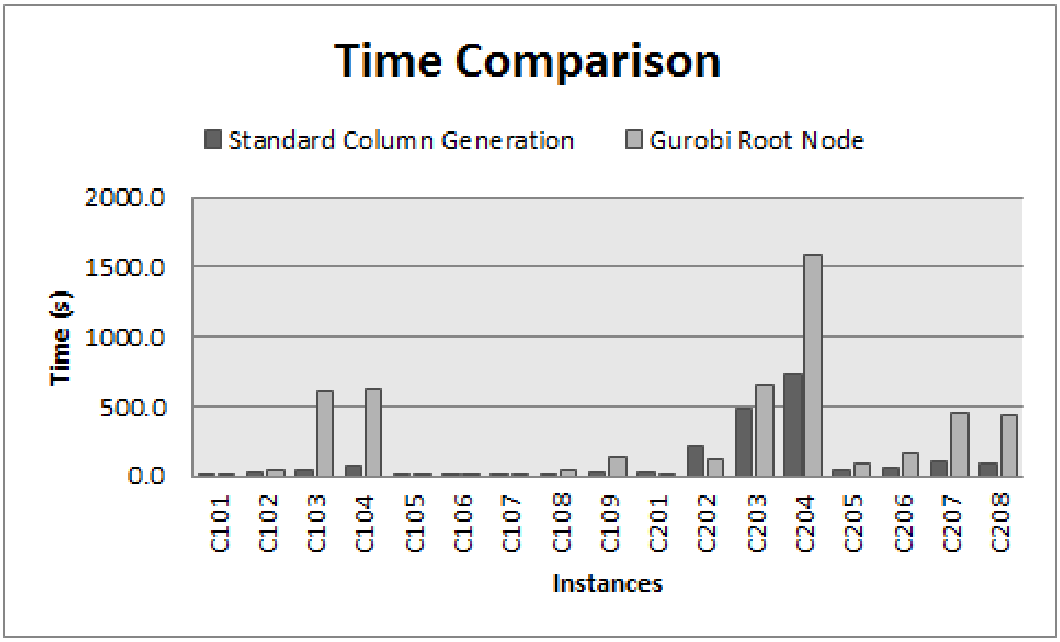

Figure VI.1:
Running Time Comparison for the 25 Customers C1 and C2 instances

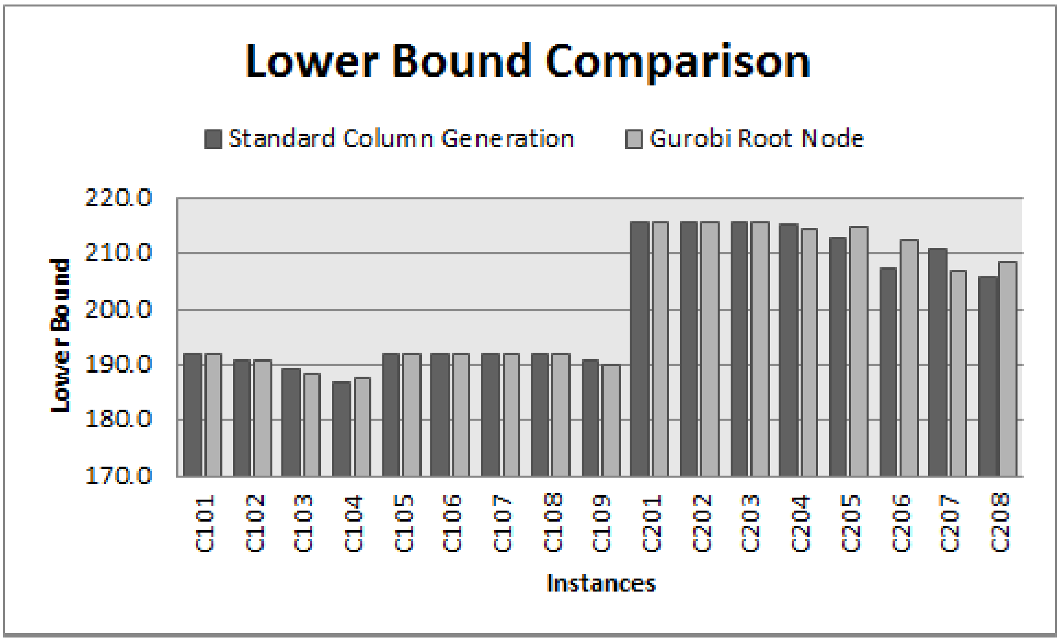

Figure VI.2:
Lower Bounds Comparison for the 25 Customers C1 and C2 instances

Table VI.2:
Results Comparison for Solomons R1 and R2 instances with 25 Customers (No Synchronization)

	Best Known		**Standard Column Generation**					**Gurobi Root Node**			
Instance	**V**	**TD**	**TD**	**Gap**	**Total Routes**	**Integer**	**Time (s)**	**TD**	**Gap**	**Integer**	**Time (s)**
R101	8	617,1	618,3	0,20%	121	Yes	0,1	618,3	0,20%	Yes	0,3
R102	7	547,1	548,1	0,18%	223	Yes	1,4	548,1	0,18%	Yes	2,6
R103	5	454,6	455,6	0,22%	262	Yes	3,0	473,4	3,97%	Yes	5,5
R104	4	416,9	418,0	0,25%	350	Yes	5,4	418,0	0,25%	Yes	9,5
R105	6	530,5	531,5	0,19%	145	Yes	0,4	531,5	0,19%	Yes	0,8
R106	3	465,4	466,5	0,23%	270	No	2,3	469,0	0,76%	Yes	4,1
R107	4	424,3	424,7	0,09%	282	No	3,8	425,3	0,23%	Yes	13,6
R108	4	397,3	398,3	0,25%	275	Yes	4,7	398,3	0,25%	Yes	24,2
R109	5	441,3	464,7	5,03%	194	Yes	1,2	462,2	4,52%	No	4,7
R110	4	444,1	440,3	0,87%	209	No	2,1	428,9	3,55%	No	13,9
R111	5	428,8	426,1	0,63%	234	No	3,0	416,3	2,99%	No	13,0
R112	4	393,0	390,9	0,55%	269	No	4,3	376,1	4,48%	No	14,3
R201	4	463,3	461,3	0,43%	949	No	14,4	464,4	0,23%	Yes	8,0
R202	4	410,5	408,0	0,61%	1.353	No	65,5	411,5	0,24%	Yes	189,6
R203	3	391,4	381,2	2,68%	1.268	No	101,8	350,0	11,84%	No	274,7
R204	2	355,0	337,1	5,31%	1.561	No	184,4	317,3	11,88%	No	864,0
R205	3	393,0	387,2	1,50%	1.244	No	38,8	378,5	3,82%	No	71,3
R206	3	374,4	369,9	1,21%	1.344	No	89,1	340,7	9,88%	No	147,2
R207	3	361,6	359,4	0,61%	1.515	No	149,2	333,6	8,40%	No	768,0
R208	1	328,2	319,2	2,81%	1.379	No	161,8	297,2	10,42%	No	1899,0
R209	2	370,7	354,9	4,45%	1.189	No	52,5	341,1	8,69%	No	207,5
R210	3	404,6	401,4	0,80%	1.266	No	74,3	357,0	13,33%	No	134,3
R211	2	350,9	335,5	4,58%	1.187	No	84,2	316,1	11,02%	No	373,0

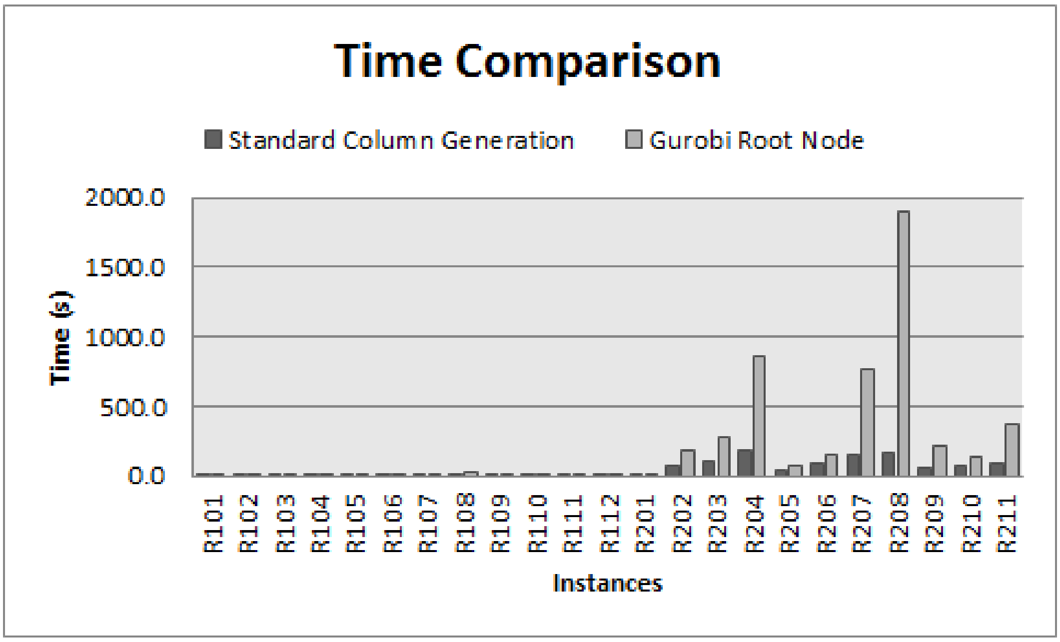

Figure VI.3:
Running Time Comparison for the 25 Customers R1 and R2 instances

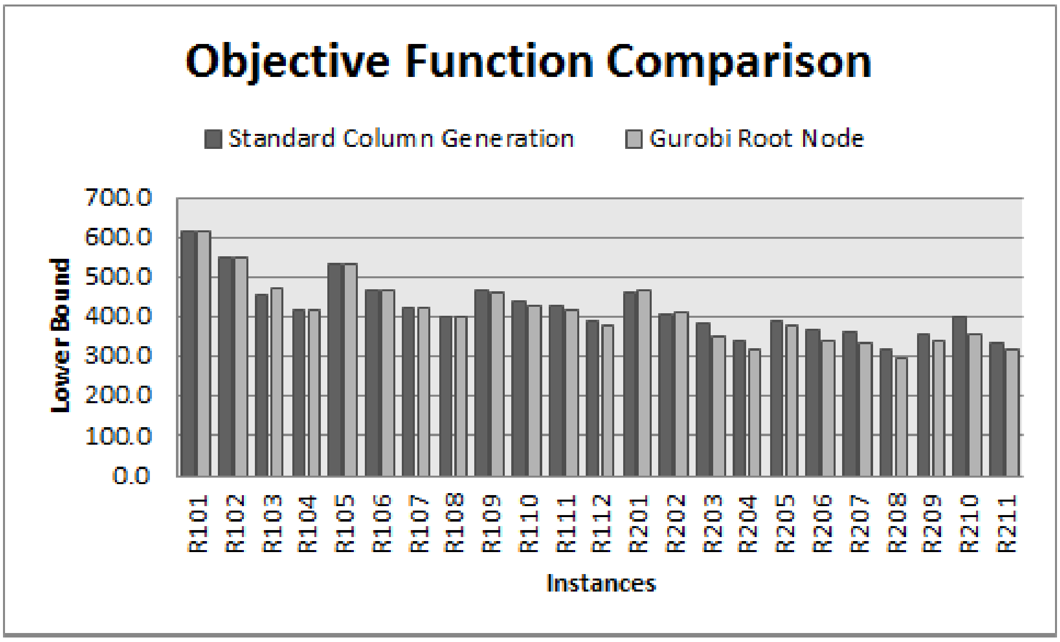

Figure VI.4:
Lower Bounds Comparison for the 25 Customers R1 and R2 instances

Table VI.3:
Results Comparison for Solomons RC1 and RC2 instances with 25 Customers (No Synchronization)

	Best Known		**Standard Column Generation**					**Gurobi Root Node**			
Instance	**V**	**TD**	**TD**	**Gap**	**Total Routes**	**Integer**	**Time (s)**	**TD**	**Gap**	**Integer**	**Time (s)**
RC101	4	461,1	426,6	8,09%	177	No	0,5	432,5	6,61%	No	1,4
RC102	3	351,8	352,9	0,32%	210	Yes	1,6	352,9	0,32%	Yes	5,1
RC103	3	332,8	333,3	0,16%	325	No	3,6	321,0	3,67%	No	14,0
RC104	3	306,6	306,3	0,09%	393	No	5,6	300,2	2,14%	No	20,3
RC105	4	411,3	412,0	0,17%	297	No	1,5	412,4	0,26%	Yes	3,1
RC106	3	345,5	342,7	0,81%	199	No	1,1	327,1	5,63%	No	5,1
RC107	3	298,3	299,0	0,22%	296	Yes	2,7	296,5	0,62%	No	13,5
RC108	3	294,5	294,2	0,09%	295	No	4,2	287,4	2,47%	No	18,5
RC201	3	360,2	370,4	2,76%	840	No	13,6	358,1	0,58%	No	14,0
RC202	3	338,0	332,9	1,54%	1.128	No	50,6	324,9	4,04%	No	197,8
RC203	3	326,9	303,5	7,71%	1.411	No	111,5	290,7	12,44%	No	338,4
RC204	3	299,7	293,1	2,24%	1.474	No	156,9	286,0	4,78%	No	611,6
RC205	3	338,0	336,7	0,38%	883	No	27,6	330,7	2,20%	No	48,8
RC206	3	324,0	319,5	1,41%	1.126	No	37,2	317,0	2,21%	No	61,0
RC207	3	298,3	296,5	0,61%	1.195	No	53,8	289,6	2,99%	No	121,6
RC208	2	267,1	222,9	19,85%	1.109	No	78,5	213,5	25,12%	No	298,0

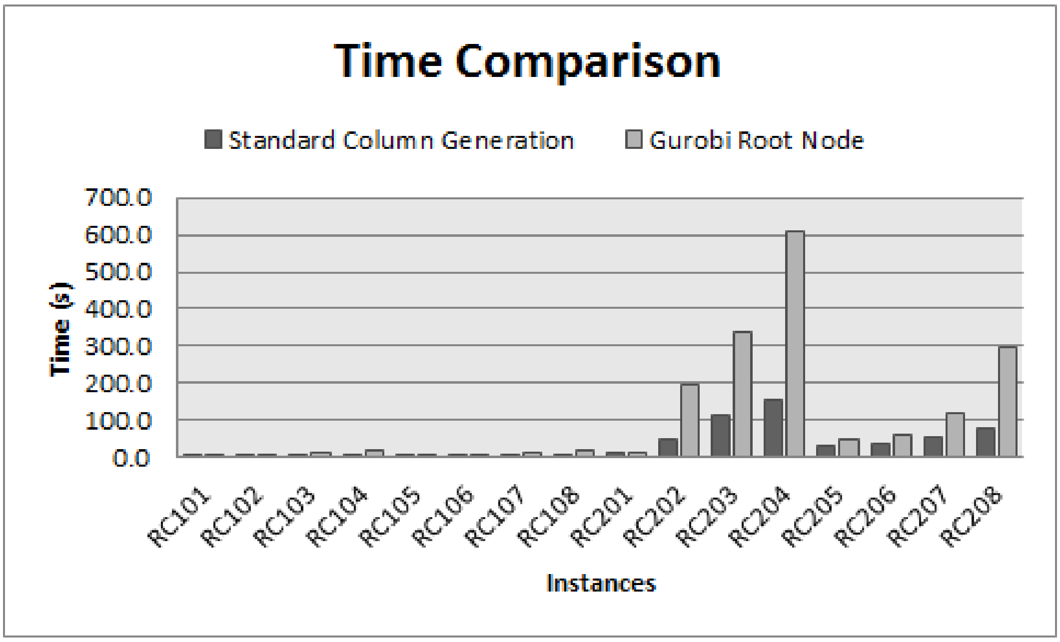

Figure VI.5:
Running Time Comparison for the 25 Customers RC1 and RC2 instances

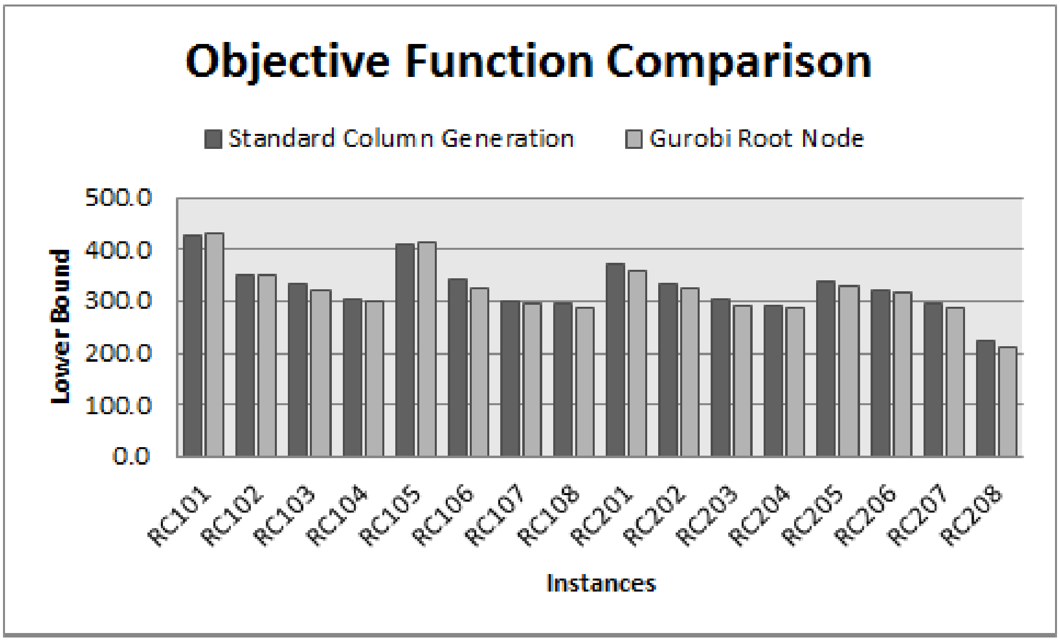

Figure VI.6:
Lower Bounds Comparison for the 25 Customers RC1 and RC2 instances

Table VI.4:
Results for Solomons instances with 50 Customers (No Synchronization)

	Best Known		Standard Column Generation				
Instance	**V**	**TD**	**TD**	**Gap**	**Total Routes**	**Integer**	**Time (s)**
C101	5	362,40	363,20	0,23%	176	Yes	4,4
C102	5	361,40	362,20	0,21%	1.457	Yes	181,5
C103	5	361,40	355,50	1,65%	1.556	No	422,1
C104	5	358,00	350,50	2,13%	1.127	No	583,3
C105	5	362,40	362,00	0,10%	256	Yes	12,4
C106	5	362,40	363,20	0,23%	507	Yes	17,7
C107	5	362,40	362,00	0,10%	1.818	Yes	127,8
C108	5	362,40	362,00	0,10%	1.260	Yes	120,6
C109	5	362,40	360,00	0,66%	975	No	161,0
C201	3	360,20	361,80	0,44%	3.189	Yes	220,8
C202	3	360,20	361,80	0,44%	5.460	Yes	2551,6
C203	3	359,80	361,40	0,45%	5.782	Yes	5609,6
C204	2	350,10	351,70	0,46%	6.820	Yes	12752,4
C205	3	359,80	360,80	0,28%	5.545	No	759,0
C206	3	359,80	354,50	1,50%	4.329	No	935,3
C207	3	359,60	358,00	0,45%	3.709	No	1321,8
C208	2	350,50	347,00	1,00%	2.408	No	737,0
R101	12	1044,00	1046,70	0,26%	330	Yes	1,0
R102	11	909,00	911,40	0,27%	675	Yes	12,1
R103	9	772,90	769,80	0,40%	621	No	24,1
R104	6	625,40	620,60	0,77%	728	No	53,4
R105	9	899,30	911,00	1,29%	427	No	3,8
R106	5	793,00	831,20	4,59%	608	No	14,9
R107	7	711,10	716,60	0,76%	649	No	28,6
R108	6	617,70	606,30	1,88%	768	No	59,1
R109	8	786,80	784,00	0,36%	431	No	9,0
R110	7	697,00	702,00	0,71%	456	No	17,1
R111	7	707,20	698,50	1,25%	568	No	25,1
R112	6	630,20	616,70	2,19%	480	No	29,6
R201	6	791,90	790,80	0,14%	2.978	No	164,6
R202	5	698,50	698,10	0,06%	3.985	No	646,9
R203	5	605,30	603,70	0,26%	4.296	No	1286,1
R204	2	506,40	481,00	5,29%	4.272	No	2434,4
R205	4	690,10	669,20	3,12%	3.054	No	375,7
R206	4	632,40	620,90	1,86%	3.969	No	925,8
R207	2	593,95	541,60	9,66%	3.329	No	1225,1
R208	2	508,41	470,20	8,12%	3.444	No	1929,5
R209	4	600,60	586,60	2,38%	3.298	No	632,2
R210	4	645,60	629,50	2,56%	3.665	No	782,7
R211	3	535,50	512,20	4,55%	2.948	No	830,6
RC101	8	944,00	874,80	7,91%	343	No	2,2
RC102	7	822,50	765,90	7,40%	472	No	7,4
RC103	6	710,90	671,00	5,94%	687	No	21,2
RC104	5	545,80	544,00	0,33%	790	No	40,3
RC105	8	855,30	813,80	5,10%	480	No	7,2
RC106	6	723,20	684,00	5,73%	410	No	6,2
RC107	6	642,70	624,60	2,90%	520	No	14,6
RC108	6	598,10	591,10	1,18%	613	No	24,6
RC201	5	684,80	684,50	0,04%	2.363	No	127,3
RC202	5	613,60	585,50	4,81%	2.912	No	406,9
RC203	4	555,30	492,10	12,84%	3.002	No	771,7
RC204	3	444,20	380,30	16,79%	3.019	No	1365,2
RC205	5	630,20	615,40	2,41%	2.823	No	294,4
RC206	5	610,00	585,80	4,13%	2.789	No	305,8
RC207	4	558,60	486,50	14,83%	2.793	No	513,6

(b) Description of the Results

The first round of experiments over the original 25 customer Solomon instances with only one vehicle type (no synchronization), showed that solving the linear relaxation of the problem with the column generation approach was, with a few exceptions, consistently faster than solving it with Gurobi. Both approaches achieved good bounds (gap usually below 1%), but it took Gurobi more time to arrive at such bounds.

One important observation that shall be done is that the lower bounds found by solving the flow formulation were some times greater than the known optimal values. As this is a minimization problem, such a behavior could be pointing out a problem. It is possible that the difference comes from the fact that the flow formulation is time indexed, i.e. time windows are discretized. This fact may induce a difference in the optimal routes encountered in the literature and the routes obtained in this work.

We will mainly use the first result table VI.1 that corresponds to the types C1 and C2 of instances with 25 customers, to make the analysis of the first round of experiments. This instances have two main characteristics: small number of vehicles and a large planning horizon. During our experiments we observed that these two factors make the problem much harder to solve.

We attribute this issue to:

- A small number of vehicles means that each vehicle has to serve more customers. Thus the column generation sub-problem tends to generate larger routes. Generated routes are q-Routes with 2-cycle elimination, thus the longer the routes the more cycles they may have, and as we already know, this is a drawback to the column generation approach.
- By the above, the column generation may have a convergence delay. This means that more routes are to be generated by an increased number of pricing calls. Recall the complexity of our pricing algorithm, which is $\mathcal{O}(n^2T)$. This means that the large planning horizons T of these instances will make the pricing algorithm expensive. Summing these two factors resumes in more calls of a more expensive algorithm.

Another interesting observation is that, as the time windows of the customers grow, the problem also becomes harder to solve. Larger time windows have the same effects, of a large time horizon, over the complexity of the pricing algorithm. Still, larger time windows introduce another inconvenient: symmetry. In this case, symmetry is caused because a lot of customers can be attended at a lot of time instants, so there may exist a lot of indistinguishable routes in the problem.

Finally, let us notice that despite the above issues, for 10 of 17 C type instances, the column generation arrived at an LP optimum that in fact was an optimal integral solution. Two possible explanations come to mind. First it is important to mention another characteristic of these instances that was not mentioned before. In these instances, the customers are dispersed in a clustered fashion. This implies that vehicles will mainly attend customers that are in the same cluster, meaning that the subset of customers a vehicle will have to visit is practically imposed a-priori. The second explanation would be that the proposed flow model and decomposition approach are promissory.

Instances C201 to C204 are good examples to identify all what was previously explained. The observations here made are consistent in all the results tables of the first round of experiments. For instance, we refer to the table with the results obtained for the 50 Customer Solomon Instances VI.4. All the CPU time increases in this table corresponds to some or all of the issues previously mentioned. A clear example of this are instances C204, R204 and RC204, which are the hardest to solve of each problem class. All of them present the above characteristics, small number of vehicles, large planing horizons (3000+ time instants), and large time windows defined for each customer. Still, once again 10 of the 17 C type clustered instances were solved to optimality by the LP.

VI.2 Results - Round 2

The second round of experiments intends to give a comparison of the difficulty brought to the VRPTW by the synchronization constraints. Such comparison is based over a practical experiment involving a set of 17 instances with and without synchronization. In order to provide fair conditions to the experiment for both types of instances, we ran this comparison using the generic Gurobi MIP solver, with default settings and time limit set to 15 minutes for solving the root relaxation and 15 minutes for the exploration of other nodes in the search tree. Results of this experiment are found in Tables VI.5 and VI.6.

A second intention of this round of experiments was to test the suggested flow model and decomposition approach in solving the VRPTWEOS. Once again, we used our adapted instances from the Solomon benchmark, and compared the results with those obtained by using just Gurobi. In these experiments, rather than using the Dantzig-Wolfe Master, we moved to the column generation for extended formulations approach V.4.

We used the column generation for extended formulations approach to solve the linear relaxation of each problem instance. As already mentioned, this approach works on the original variable flow formulation, which at the beginning (in our implementation) has none of the x variables. All other variables remain in the model but as continuous variables. At the end of the algorithm, when a solution for the linear relaxation is found, we arrive at a model with only a portion of the initial defined set of x variables.

At this point, if the obtained solution is an integer solution, the algorithm stops, as this solution is optimal. Otherwise, another method must be used in order to find integer solutions. With this in mind, we tested the branch and price algorithm described in Section V.5(a). However, we found that it performed poorly for the instances with synchronization, sometimes even failing to find a first integer solution. Still, we were able to use our branch and price algorithm to support the method for adding synchronization requirements to the Solomon instances, described in Section VI.2(a).

Because of this observation, we decided to test a simple approximation algorithm, which we now describe. As mentioned before, after applying the column generation for extended formulations approach to solve the linear relaxation of the flow formulation, we arrive at a model with only a portion of the initial defined set of x variables. We take that resulting model and reinstate the integrality constraints. Finally, this smaller integer programming model is solved by Gurobi. Results of this approach are shown in Table VI.7.

This section is organized as follows. Firstly, we describe the method used to include synchronization requirements in some of the Solomon instances. Next, Tables VI.5 to VI.7 and Figures VI.7 and VI.8 show the results obtained from the experiments mentioned above. Finally a review of those results is presented.

(a) Adding Synchronization to VRPTW Instances

To evaluate the level of difficulty introduced to the VRPTW by the synchronization constraints, we created a set of 17 instances based on a small group of instances of the Solomon benchmark, the base set of instances from now on. The method used to add synchronization to those instances is straightforward, and aims to build instances which feasibility is guaranteed. A description of this method follows.

Consider the set of routes presented below that constitutes an integer solution to a VRPTW instance with 25 customers. In terms of the sequence of visited customers, these routes are:

- *Route 1: 0 5 3 7 8 10 11 9 6 4 2 1 0*
- *Route 2: 0 13 17 18 19 15 16 14 12 0*
- *Route 3: 0 20 24 25 23 22 21 0*

As stated in Section III.2, the VRPTWEOS can be seen as a generalization of the VRPTW, which implies that the above is also a solution to the same instance of a VRPTWEOS with only one vehicle type e_1.

By definition, synchronization involves more than one type of vehicle. Hence, our first step is adding to this instance two more vehicle types e_2 and e_3. The second step is to create the synchronization requirement for each customer, for instance by making the following association:

- *All customers visited in Route 1 require vehicle types e_1 and e_2.*
- *All customers visited in Route 2 require vehicle types e_1 and e_3.*
- *All customers visited in Route 3 require vehicle types e_2 and e_3.*

The third and last step is to make available 2 vehicles of each type $v_{1,e_1}, v_{2,e_1}, v_{1,e_2}, v_{2,e_2}, v_{1,e_3}, v_{2,e_3}$.

Finally, notice that a feasible solution to this synchronization instance is ensured by the following assignment of vehicles to routes:

- *All customers in Route 1 will be served by vehicles v_{1,e_1} and v_{1,e_2}*
- *All customers in Route 2 will be served by vehicles v_{2,e_1} and v_{1,e_3}*
- *All customers in Route 3 will be served by vehicles v_{2,e_2} and v_{2,e_3}*

(b) Results Tables

The headers of the result tables of the first round of experiments refer to:

- The header **Instance** refers to the name of the adapted Solomon instance.
- The header **Root Relax** corresponds to the lower bound found by solving the linear relaxation of the problem.
- The header **LB** refers to the last lower bound found during Gurobi's branch and cut algorithm.
- The header **LP Routes** is the number of routes generated by the column generation pricing algorithm, while solving the linear relaxation of the problem.
- The header **LP Time** refers to the time consumed to solve the linear relaxation of the problem.
- Header **UB** is the value of the best integer solution found, if such a solution was found.
- **Gap** header corresponds to the optimality gap between the best current integer solution and the problemś Lb.
- **Nodes** refer to the amount of nodes that Gurobi explored while solving the Integer Program.
- Table header **Variables** indicate how many variables the final MIP model solved by Gurobi had.
- Header **Total Time** refers to the total running time in seconds of the solution approach.

Table VI.5:
Results for the Base set of Solomon Instances with one type of vehicle (No synchronization)
Gurobi MIP Solver

Instance	LB	LP Routes	LP Time (s)	UB	Gap	Nodes	Variables	Total Time
C101	191,81	–	1,45	191,81	0,00%	1	31.056	1,48
C102	190,74	–	28,50	190,74	0,00%	0	130.930	37,61
C103	188,42	–	900,10	191,13	1,42%	157	273.044	1802,61
C105	191,81	–	4,03	191,81	0,00%	1	52.679	4,06
C106	191,81	–	1,88	191,81	0,00%	1	33.944	1,91
C107	191,81	–	6,32	191,81	0,00%	1	74.433	6,40
C108	191,81	–	69,63	191,81	0,00%	0	102.982	69,80
C109	190,38	–	285,40	191,81	0,00%	915	157.816	542,72
RC102	352,94	–	2,67	352,94	0,00%	0	21.505	2,74
RC103	326,45	–	26,70	334,12	0,00%	1236	35.851	58,78
RC104	301,35	–	25,78	307,14	0,00%	2530	47.382	244,49
RC106	324,41	–	13,64	346,51	0,00%	39672	18.926	517,15
RC107	296,71	–	21,98	298,95	0,00%	23	30.614	22,95
RC108	288,15	–	45,92	294,99	0,87%	49611	44.967	945,98
RC201	351,64	–	13,35	361,24	0,00%	1101	54.755	60,89
RC205	332,55	–	174,03	338,93	0,00%	66515	111.998	839,95
RC207	291,53	–	106,94	298,95	1,27%	2008	157.602	1007,30

Table VI.6:
Results for Adapted Solomon Instances with 3 Types of Vehicle
Gurobi MIP Solver

Instance	**Root Relax**	**Root Routes**	**Root Time (s)**	**LB**	**UB**	**Gap**	**Nodes**	**Variables**	**Total Time (s)**
C101	383,63	–	2,65	383,63	383,63	0,00%	1	49.850	2,70
C102	378,93	–	631,69	381,48	381,48	0,00%	1	197.154	631,79
C103	381,31	–	900,08	381,92			1	382.210	1803,90
C105	383,63	–	7,08	383,63	383,63	0,00%	1	79.799	7,15
C106	383,63	–	3,43	383,63	383,63	0,00%	1	54.107	3,47
C107	383,63	–	12,03	383,63	383,63	0,00%	1	109.583	12,10
C108	381,99	–	87,26	383,63	383,63	0,00%	0	148.631	97,80
C109	380,56	–	900,03	381,43	385,34	1,02%	18	225.751	1802,40
RC102	695,58	–	7,35	705,88	705,88	0,00%	0	33.460	7,38
RC103	635,52	–	74,57	649,45	671,04	3,22%	1223	53.496	974,61
RC104	597,75	–	228,79	614,28	614,28	0,00%	1190	69.508	793,16
RC106	640,41	–	28,47	658,17	695,38	5,35%	3222	29.566	928,48
RC107	584,80	–	143,88	592,54	597,90	0,90%	1289	46.441	1043,93
RC108	570,38	–	231,68	572,66	643,07	10,95%	153	67.119	1131,92
RC201	681,13	–	84,84	712,65	725,30	1,74%	1608	81.502	984,92
RC205	638,49	–	181,19	644,85			889	159.726	1081,23
RC207	574,82	–	863,59	582,51	602,53	3,32%	1060	220.782	1763,69

Table VI.7:
Results for Adapted Solomon Instances with 3 Types of Vehicles
Column Generation for Extended Formulations + Gurobi MIP Solver

Instance	**Root Relax**	**Root Routes**	**Root Time (s)**	**LB**	**UB**	**Gap**	**Nodes**	**Variables**	**Total Time (s)**
C101	383,63	3.409	39,05	383,63	383,63	0,00%	0	27.226	39,56
C102	378,93	20.028	621,33	381,48	381,48	0,00%	0	52.401	629,04
C103	381,31	25.737	1838,10	386,94	386,94	0,00%	475	80.545	2092,91
C105	383,63	7.014	101,77	383,63	383,63	0,00%	0	33.269	102,63
C106	383,63	3.619	44,44	383,63	383,63	0,00%	0	27.952	45,02
C107	383,63	11.353	230,36	383,63	383,63	0,00%	0	40.363	231,98
C108	381,99	19.176	578,71	383,63	383,63	0,00%	0	49.089	584,99
C109	380,56	25.566	1395,03	385,34	385,34	0,00%	334	62.773	1548,87
RC102	695,58	5.473	27,62	705,88	705,88	0,00%	0	11.635	29,25
RC103	635,42	6.635	51,83	673,47	673,47	0,00%	1504	15.435	126,14
RC104	597,92	8.230	93,31	614,28	614,28	0,00%	10	18.882	119,07
RC106	640,45	5.101	26,64	708,26	708,26	0,00%	3429	11.426	143,87
RC107	584,80	5.606	43,19	597,90	597,90	0,00%	394	14.693	62,70
RC108	571,49	6.505	77,29	576,56	589,99	2,28%	2904	18.186	977,42
RC201	692,49	7.824	119,90	731,77	731,77	0,00%	1128	32.768	128,82
RC205	638,67	11.630	320,39	677,86	677,86	0,00%	4914	42.906	918,58
RC207	575,30	17.868	724,10	603,73	603,73	0,00%	6495	55.695	1490,31

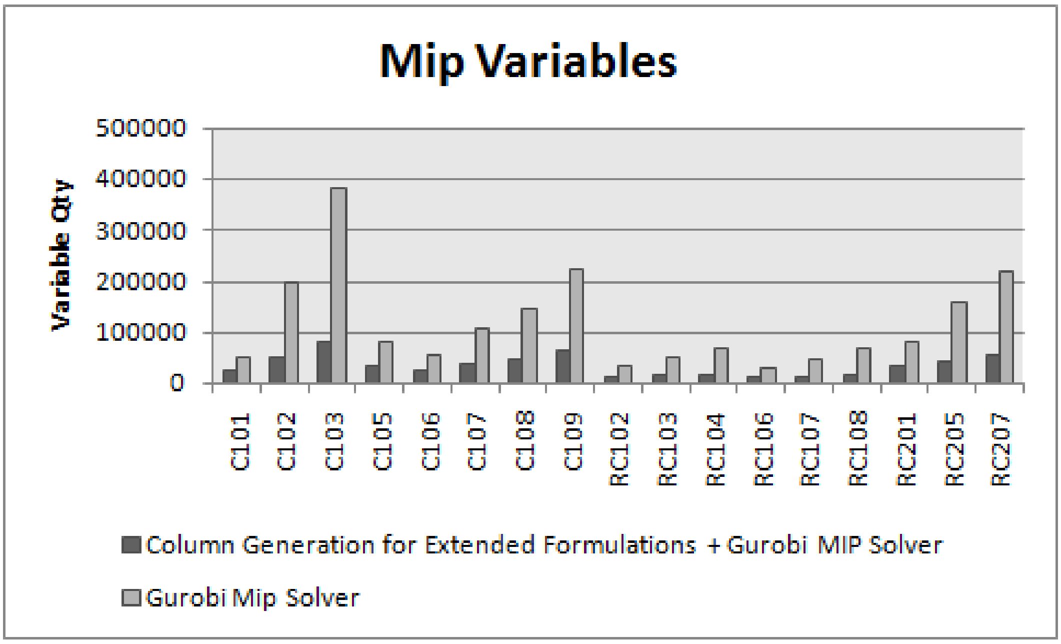

Figure VI.7:
Comparison of final number of variables in the MIP Model.
Gurobi vs Column Generation for Extended Formulations

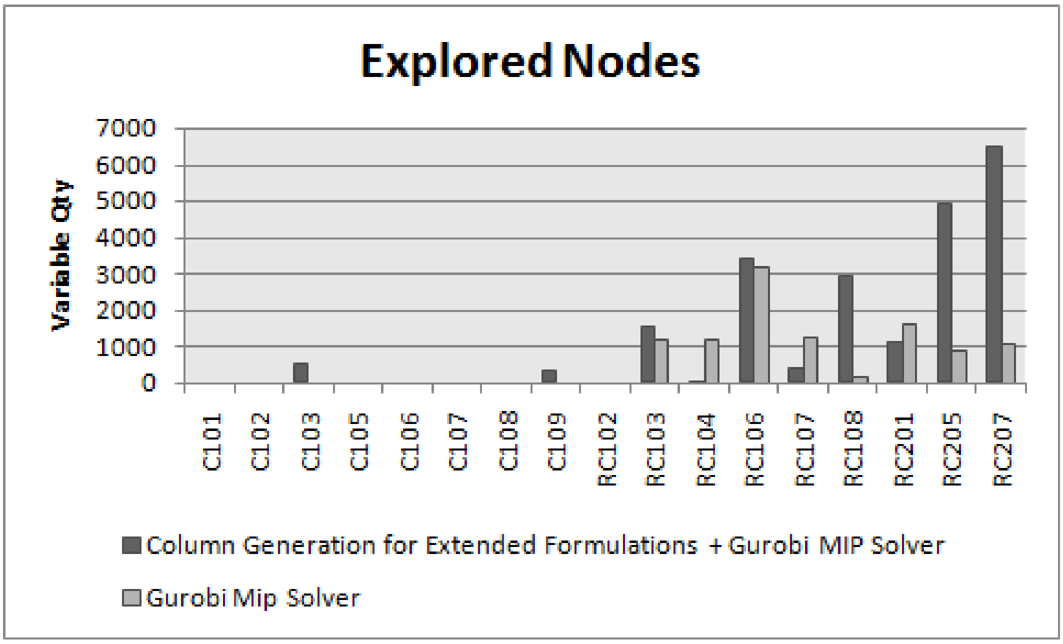

Figure VI.8:
Comparison of explored nodes of the branch-and-bound tree.
Gurobi vs Column Generation for Extended Formulations

(c) Description of the Results

The second round of experiments over the set of 17 Solomon instances, with and without synchronization showed that, at least in what concerns to the time indexed flow formulation addressed in this dissertation, the addition of synchronization instances makes the problem harder to solve.

Such a difference can be viewed in tables VI.5 and table VI.6. For example, Gurobi achieved a Gap of 1.4%, within the given running time limitations, for the C103 instance without synchronization. By contrast, for the same instance but including synchronization and three types of vehicles, Gurobi was not able to even get to a first integer solution. The increased number of vehicles is not necessarily a complicating factor, as was mentioned earlier in this chapter. However, the fact that vehicles are of different types and that each customer requires two types of vehicles, make the routes of the vehicles highly interdependent. We attribute the increase in complexity to this interdependency problem. Another symptom of this increased level of difficulty, is that the time spent by Gurobi to solve the linear relaxation of the problem increased by a factor sometimes greater than 10 (Instance C102). Therefore, fewer nodes were able to be explored in the tree and thus fewer optimal integer solutions were found.

Table VI.7 corresponds to the results of solving the same set of instances with synchronization, as in table VI.6, but by using the column generation for extended formulations approach, previously described in V.4. First, let us mention that the results obtained in this experiments reaffirms our observation that synchronization constraints difficult the VRPTW. This can be easily viewed by comparing the results here obtained to the results obtained in the first round of experiments. Table VI.1 shows results obtained by column generation for instances without synchronization. It's easy to see that the number of routes generated for the problem with synchronization constraints is much greater that the number of routes generated in the absence of synchronization. One of several examples is the instance C103, for which 572 routes where needed to solve the problem without synchronization. With synchronization, 25000+ routes needed to be generated for the same problem. The number of vehicles available for this instance increased from 3 to 6 vehicles, but still the number of routes generated increased in a much greater factor. Again, we attribute this issue to the interdependency problem. As routes are generated independently for each vehicle type, i.e. without concerning with synchronization, we observed that in practice a lot of routes had to be generated in order to get to a set of routes suitable to being synchronized (coupled) by the master problem.

From tables VI.7 and table VI.6 can be observed that, our algorithm is a promising alternative to the solely use of Gurobi. It is able to arrive at optimal or near optimal bounds in most of the instances, within the given time limitations. Nevertheless, for smaller instances Gurobi reach optimality much faster. This is the case for instances with less than 80 thousand variables. However, when the problem becomes harder to solve, i.e. with greater time windows, the pure gurobi approach suffers the most.

Another key observation is that, after solving the linear relaxation of the problem instances, the algorithm arrives to a version of the original variable formulation with much fewer variables than the complete original model. Figure VI.7 shows that this difference in the number of variables is significant and for some instances it allows further exploration of the branch-and-bound tree, see figure VI.8. Still, in some cases, e.g. instance RC106, when comparing the best bounds obtained by Gurobi and by our proposed algorithm, it can be concluded that our approach cutted out the optimal solution. This was somewhat expected, as the proposed algorithm is an approximation algorithm and not an exact algorithm.