VI
A Branch-Cut-and-Price Algorithm

As mentioned in the former chapters, since the column generation algorithm solves the linear relaxation of the set partitioning formulation, another technique is needed in order to obtain an integer solution. In this chapter, we use Branch-and-Bound, a widely known technique, which enumerates all possible solutions through a search tree, discarding the branches with solutions greater than a known upper bound for the instance. This technique, when used together with column generation and cut separation is called *Branch-Cut-and-Price* (BCP). We devise a Branch-Cut-and-Price algorithm for the CARP using the column generation algorithm described in Chapter V and exactly separating the odd edge cutset and capacity cuts as described in Chapter IV.

VI.1 The Branch-and-Bound

The BCP algorithm starts by solving the relaxation problem defined at root node of the search tree. This operation consists in executing the column generation algorithm as described in Chapter V, which includes the exact separation of cuts. If the solution is integral, i.e., all variables are integers, the algorithm stops and returns this solution. Otherwise, this node is inserted into a list and the main loop of the BCP begins. At each iteration, the algorithm chooses from the list the next node to be opened. This choice of a node can be implemented in different ways, each one reflecting in a different search through the tree. For instance, if the list is a stack, the search will be a depth-first search. In our implementation, a priority queue is used, always returning the node with the lowest value. This approach is usually called best-first search and, each time a new node is chosen from the list, its value is considered the new current lower bound. This can be assured by the fact that there is no node with a lower value remaining in the list.

The opening of a node consists in selecting a fractional variable of its solution and creating two branches, following a defined branching rule, which will be explained further. Then, from each branch a new node is generated,

which will be solved in a similar manner as the root node. However, differently from the root node, the value is tested against the known upper bound and, if it is greater or equal than the bound, it is discarded. This operation is usually known as pruning the node of the search tree. On the other hand, if the value is less than the upper bound, the node is considered as a valid node. Furthermore, if the solution of a valid node is integral, its value is then considered as the new upper bound and there is no way to branch from this node anymore. When this is not the case and the solution of a valid node still contains any fractional variable, the node is inserted into the list.

It is important to notice that, at a given iteration, if the chosen node has a value that differs less than one unit from the known upper bound, the BCP algorithm stops and the solution which generated the upper bound is then returned as the overall best solution.

(a) Branching Rules

The most intuitive branching rule is, for a given variable λ_r from the set partitioning formulation with a fractional value $\bar{\lambda_r} \in [0, 1]$, create two branches, the first one with $\lambda_r = 0$ and other with $\lambda_r = 1$. However, this cannot be done due to the column generation algorithm. It is easy to see that if the variable associated with a route r is fixed to zero, eventually the pricing subproblem will find this route again on a subsequent iteration. This route will be returned to the restricted master which will consider it as a new route and will create a new variable λ_r for it.

There are some ways to cope with this difficulty. But instead of that, we prefer to branch on the deadheaded edges variables z_e from the one-index formulation. By the time it is necessary to branch, the values of the z_e variables are obtained using the relation $\sum_{r\in\Omega} b_r^e \lambda_r = z_e, \forall e \in E$, which is analogous to the one created for the two-index formulation, shown in (III.20).

The drawback of this solution is that when an integer solution is found, it is integer just on z_e variables and it may not be integer on the λ_r variables. This integer solution has the same properties of one from the one-index formulation, it may have a value less than the optimal solution, which would also be infeasible. However, it does not prevent the BCP algorithm from returning good solutions, as will be shown in the results.

The selection of the branching variable can follow different rules. The simplest rule is to search for the variable whose value in the solution $\bar{z}_e$ is closer to 0.5 and then create two branches, one with $z_e \leq \lfloor \bar{z}_e \rfloor$ and other with $z_e \geq \lceil \bar{z}_e \rceil$. Mapping these inequalities to λ_r variables results in the following two constraints.

$$\sum_{r\in\Omega} b_r^e \lambda_r \leq \lfloor \bar{z}_e \rfloor \tag{VI.1}$$

$$\sum_{r\in\Omega} b_r^e \lambda_r \geq \lceil \bar{z}_e \rceil \ . \tag{VI.2}$$

Instead of creating these two constraints for each new branch generated during the BCP algorithm, one bound constraint for each deadheaded edge $e \in E$ can be inserted *a priori* in the restricted master as shown below.

$$lb_e \leq \sum_{r\in\Omega} b_r^e \lambda_r \leq ub_e \quad \forall e \in E \ . \tag{VI.3}$$

With the introduction of the bounds constraints, the pricing subproblem does not change, but the reduced cost of an edge, shown in equations (V.2) and (V.3), must consider the dual values associated with these constraints. Therefore, given ρ_e, the dual variable associated with constraints (VI.3), this value must be subtracted from the reduced cost of the associated edge e for both equations.

In this work, the branching rule described above is used. However, there are some different branching rules which could be used. For instance, instead of choosing the variable closer to 0.5, the rule could be to choose the variable closer to any other value between 0 and 1, exclusive. Furthermore, a set of edges could be chosen. One way to do this is to select a vertex set S with $\sum_{e\in\delta(S)} z_e$ fractional and create the associated branches.

VI.2 Improving the Search

(a) Using Strong Branching

Strong Branching is a technique used in the attempt to find better lower bounds faster. Instead of choosing the best fractional variable w.r.t. the branching rule, it uses a candidate set of variables and tests which of them give the best progress before actually perform any branching. This test is done by solving the linear relaxations – in this case the column generation – of the two nodes which would be created from each candidate. If the candidate set is the full set of fractional variables and, for each candidate, both linear relaxations are solved to optimality, this strategy is called *full strong branching*. A benefit of this strategy is to always choose the locally best fractional variable to branch on. However, the effort needed for solving each node of the search tree is high. Alternatively, most strong branchings in literature tries to estimate quickly which variable is a good one to branch on.

One possibility to speed up the strong branching is to evaluate only a subset of the candidate set. In this work, we create the candidate set by choosing the best 5 fractional variables w.r.t. the branching rule. For each candidate s, an estimate on the values of both the linear relaxations is obtained by running the column generation using just the heuristic pricing described on Section V.1(c) and without separating any cut. Let $left_s$ and $right_s$ be these values. In order to choose which candidate is the one to branch, an evaluation criterion is used. The most common one is to select the one with largest $min\{left_s, right_s\}$ and, in the case of a tie, the one with largest $max\{left_s, right_s\}$. We use a more complex criterion, as follows, which usually gives better results than this simple one.

$$\sigma(s) = \alpha \min(left_s, right_s) + (1 - \alpha) \max(left_s, right_s) \tag{VI.4}$$

This function can be calibrated using different values for the parameter $\alpha \in [0, 1]$. During our experiments, we found $\alpha = 3/4$ to be a good choice. Therefore, we branch on the candidate with largest $\sigma(s)$.

As mentioned, during the strong branching evaluation the linear relaxations are not solved to optimality. This approach is called strong branching on pseudo-costs, since we do not know the optimal value of the linear relaxations. This was originally proposed for problems which do not use column generation. In this case, the pseudo-costs are obtained performing only a few iterations of the linear programming solver and these values are used to choose the branching variable. Anyway, in both cases, when the branching variable is finally chosen, the optimal values must be computed. This requires our algorithm to do a complete execution of the column generation for both new nodes of the search tree.

(b) Using Reduced Cost Fixing

Reduced Cost Fixing is a well-known technique which is used to estimate the limits of a variable using its current reduced cost and the known lower and upper bounds. We refer the reader to the 1998 book of Wolsey [79] where a detailed description of its applications is given. The benefit of using this technique is the possible reduction in the number of nodes during the branch-and-bound search if some variable fixings are done at each iteration of the algorithm.

Given a variable z_e with reduced cost $\bar{c}_e$ and solution value $\tilde{z}_e = lb_e$ equals to its current lower bound, it is correct to state that if the value of this variable is incremented by 1, the value of the current solution is incremented

by $\bar{c}_e$. Therefore, knowing the value of the current solution Z^* and an upper bound UB for the solution, it can be concluded that an upper bound for this variable is as follows.

$$ub'_e = lb_e + \left\lfloor \frac{UB - Z^*}{\bar{c}_e} \right\rfloor \tag{VI.5}$$

The upper bound for this variable is updated in case $ub'_e < ub_e$. Analogously, given a variable z_e with solution value $\tilde{z}_e = ub_e$ equals to its current upper bound and a known lower bound LB for the solution, it can also be concluded that a lower bound for the variable is as follows.

$$lb'_e = ub_e - \left\lceil \frac{UB - Z^*}{|\bar{c}_e|} \right\rceil \tag{VI.6}$$

Notice that these bounds can only be computed for variables with non-negative reduced costs $\bar{c}_e \neq 0$, otherwise it would result in a division by zero.

VI.3 Experimental Results

The computational experiments were done using the same configuration as Chapters IV and V. The BCP algorithm was tested on all CARP datasets described on section IV.4, except for the *egl-large* dataset. As mentioned before, the instances of this dataset are prohibitively large to column generation.

The results are shown in Table VI.1. Columns `Ins`, `LB` and `UB` show the name and the known lower and upper bounds for each instance. Following these columns, for each group of columns `NG=X`, where $\Delta(N_i) = X$, columns `Value`, `Routes`, `Cuts`, `Nodes` and `Time` show the solution cost, the number of routes found by the column generation, the number of cuts found by the separation algorithms, the number of nodes opened by the branch-and-bound and the overall time of computation in seconds, respectively. For each instance, a time limit of three hours was used. Furthermore, when a solution cost is better than the known lower bound, its value is highlighted in boldface, and when it is an optimal solution for the instance, it is underlined.

Notice that even with the small time limit used and for small values of $\Delta(N_i)$, the algorithm was able to improve several known lower bounds and it was also able to find some new optimal values for the beullens dataset.

Table VI.1: Branch-Cut-and-Price Results for the CARP

Ins	LB	UB	NG=8					NG=16					NG=32				
			Value	Routes	Cuts	Nodes	Time	Value	Routes	Cuts	Nodes	Time	Value	Routes	Cuts	Nodes	Time
kshs1	14661	14661	14661.0	116	5	1	0.110	14661.0	92	5	1	0.109	14661.0	92	5	1	0.094
kshs2	9863	9863	9863.0	124	8	1	0.109	9863.0	88	8	1	0.109	9863.0	88	8	1	0.078
kshs3	9320	9320	9320.0	167	4	1	0.141	9320.0	93	4	1	0.125	9320.0	93	4	1	0.125
kshs4	11498	11498	11498.0	134	5	1	0.156	11498.0	99	5	1	0.141	11498.0	99	5	1	0.188
kshs5	10957	10957	10957.0	110	4	1	0.156	10957.0	112	4	1	0.219	10957.0	112	4	1	0.203
kshs6	10197	10197	10197.0	234	4	1	0.172	10197.0	123	4	1	0.109	10197.0	123	4	1	0.188
gdb1	316	316	316.0	259	12	1	0.031	316.0	235	12	1	0.016	316.0	201	12	1	0.031
gdb2	339	339	339.0	260	8	1	0.031	339.0	303	8	1	0.031	339.0	290	8	1	0.031
gdb3	275	275	275.0	217	5	1	0.016	275.0	200	5	1	0.015	275.0	191	5	1	0.016
gdb4	287	287	287.0	126	6	1	0.000	287.0	86	6	1	0.000	287.0	86	6	1	0.016
gdb5	377	377	377.0	228	8	1	0.032	377.0	200	8	1	0.015	377.0	197	8	1	0.031
gdb6	298	298	298.0	158	5	1	0.015	298.0	160	4	1	0.016	298.0	143	4	1	0.016
gdb7	325	325	325.0	219	10	1	0.016	325.0	147	10	1	0.016	325.0	192	10	1	0.015
gdb8	348	348	347.0	944	28	13	1.688	347.0	828	24	11	1.266	347.0	885	25	13	1.672
gdb9	303	303	303.0	813	18	7	1.469	303.0	734	17	3	0.844	303.0	857	19	3	0.922
gdb10	275	275	275.0	245	12	1	0.032	275.0	228	12	1	0.031	275.0	237	13	1	0.047
gdb11	395	395	395.0	1105	13	1	0.953	395.0	925	13	1	0.782	395.0	1105	13	1	0.968
gdb12	458	458	454.0	253	11	5	0.203	455.0	236	11	3	0.156	455.0	202	11	3	0.141
gdb13	536	536	536.0	412	5	1	0.172	536.0	400	5	1	0.172	536.0	291	5	1	0.125
gdb14	100	100	100.0	151	1	1	0.016	100.0	165	1	1	0.031	100.0	135	1	1	0.015
gdb15	58	58	58.0	127	1	3	0.062	58.0	174	1	3	0.109	58.0	220	1	5	0.157
gdb16	127	127	127.0	378	6	3	0.125	127.0	322	6	3	0.110	127.0	228	7	3	0.093
gdb17	91	91	91.0	134	7	1	0.047	91.0	134	7	1	0.093	91.0	74	7	1	0.032
gdb18	164	164	164.0	565	1	1	0.219	164.0	685	1	1	0.297	164.0	385	1	1	0.156
gdb19	55	55	55.0	55	4	1	0.016	55.0	60	4	1	0.015	55.0	60	4	1	0.000
gdb20	121	121	121.0	265	9	1	0.047	121.0	205	10	1	0.063	121.0	187	10	1	0.047

Continued on next page

Table VI.1: *Continued from previous page*

Ins	LB	UB	NG=8					NG=16					NG=32				
			Value	Routes	Cuts	Nodes	Time	Value	Routes	Cuts	Nodes	Time	Value	Routes	Cuts	Nodes	Time
gdb21	156	156	156.0	454	7	3	0.219	156.0	354	6	3	0.281	156.0	314	7	3	0.172
gdb22	200	200	200.0	452	7	11	0.875	200.0	472	7	9	0.890	200.0	412	6	11	1.015
gdb23	233	233	233.0	412	1	17	1.484	233.0	432	1	7	0.703	233.0	352	1	9	0.859
1A	173	173	173.0	3067	20	3	8.516	173.0	3192	20	3	10.812	173.0	2754	20	3	10.750
1B	173	173	173.0	1762	20	1	2.985	173.0	1711	20	3	4.094	173.0	1809	20	3	4.641
1C	245	245	242.0	1094	28	31	5.750	242.0	696	18	5	1.484	242.0	713	18	5	1.672
2A	227	227	227.0	2107	12	1	4.063	227.0	2138	11	1	5.063	227.0	1250	11	1	6.109
2B	259	259	258.0	2662	25	5	7.515	258.0	2522	25	3	16.141	258.0	1708	25	1	8.219
2C	457	457	457.0	406	16	1	0.328	457.0	398	16	1	0.297	457.0	345	16	1	0.265
3A	81	81	81.0	2589	25	3	4.282	81.0	2867	23	3	5.969	81.0	2340	24	3	5.938
3B	87	87	87.0	2591	35	7	5.344	87.0	2755	36	7	5.797	87.0	2670	40	9	9.594
3C	138	138	136.0	764	13	9	0.750	136.0	792	15	9	0.953	136.0	775	17	9	0.921
4A	400	400	400.0	9283	37	3	110.140	400.0	10281	37	3	141.953	400.0	12046	37	3	247.156
4B	412	412	412.0	4576	34	3	29.515	412.0	5996	36	3	49.375	412.0	6596	34	3	64.203
4C	428	428	428.0	4297	33	3	26.390	428.0	4771	33	3	30.844	428.0	3846	33	3	28.406
4D	530	530	527.0	5302	75	85	149.984	528.0	10580	102	371	1224.310	528.0	6431	78	95	303.688
5A	423	423	423.0	4487	25	3	31.297	423.0	5547	26	3	42.781	423.0	5327	26	3	44.687
5B	446	446	446.0	4901	73	11	60.890	446.0	6615	125	11	102.406	446.0	7771	74	11	129.906
5C	474	474	471.0	8552	119	25	196.500	471.0	6127	127	13	97.890	471.0	8893	125	19	200.469
5D	577	577	573.0	4734	166	27	61.219	573.0	2564	110	11	20.532	573.0	2414	111	9	21.547
6A	223	223	223.0	3475	21	3	13.797	223.0	3849	21	3	16.907	223.0	3458	22	3	14.969
6B	233	233	231.0	6120	86	37	73.422	231.0	6068	125	19	75.266	231.0	2961	79	7	18.718
6C	317	317	313.0	1198	48	25	8.797	313.0	1623	46	111	45.422	313.0	1344	46	57	23.328
7A	279	279	279.0	4507	23	1	23.235	279.0	4019	21	1	24.547	279.0	5376	21	1	40.532
7B	283	283	283.0	3616	21	1	14.156	283.0	3576	21	1	17.563	283.0	3815	21	1	19.469
7C	334	334	330.0	6057	135	71	114.078	330.0	4987	133	47	96.735	330.0	4125	109	37	78.453
8A	386	386	386.0	3167	23	1	15.563	386.0	3127	23	1	16.531	386.0	3567	23	1	21.515

Continued on next page

Table VI.1: *Continued from previous page*

Ins	LB	UB	NG=8					NG=16					NG=32				
			Value	Routes	Cuts	Nodes	Time	Value	Routes	Cuts	Nodes	Time	Value	Routes	Cuts	Nodes	Time
8B	395	395	395.0	2496	24	1	9.609	395.0	3116	21	1	13.140	395.0	3216	22	1	15.000
8C	521	521	518.0	2715	71	23	23.735	518.0	2506	68	23	23.796	518.0	2108	60	15	15.750
9A	323	323	323.0	9967	54	3	173.109	323.0	11886	54	3	263.953	323.0	11746	54	3	303.500
9B	326	326	326.0	9956	49	3	149.875	326.0	11336	49	3	228.532	326.0	16736	50	3	701.781
9C	332	332	332.0	6025	44	5	64.593	332.0	7765	44	5	107.969	332.0	6485	45	3	77.156
9D	391	391	386.6	24915	753	545	*3h*	387.0	22472	599	429	6646.190	387.0	15283	438	201	2414.980
10A	428	428	428.0	10749	44	3	241.000	428.0	16333	44	3	640.140	428.0	21067	44	3	1516.860
10B	436	436	436.0	8756	59	5	183.438	436.0	9116	59	5	236.032	436.0	10176	58	5	304.297
10C	446	446	446.0	8563	60	9	209.859	446.0	8285	65	7	176.375	446.0	9752	63	7	245.609
10D	526	526	525.0	8806	132	33	260.328	525.0	7110	99	21	140.906	525.0	9474	154	31	303.156
C01	4105	4150	**4125.0**	33153	470	471	6974.130	**4125.0**	20467	345	265	2994.160	**4125.0**	18158	328	209	2917.990
C02	3135	3135	3135.0	2081	32	1	3.453	3135.0	2108	32	3	5.281	3135.0	2040	32	3	6.016
C03	2575	2575	2565.0	5145	113	23	48.016	2565.0	4668	125	27	52.515	2565.0	3938	134	17	45.219
C04	3478	3510	**3495.0**	7310	76	11	55.188	**3500.0**	8775	82	21	127.375	**3500.0**	7333	81	13	125.000
C05	5365	5365	5335.0	5022	90	27	64.078	5335.0	4266	73	23	46.547	5340.0	6215	119	73	179.031
C06	2535	2535	2525.0	6446	76	69	89.735	2525.0	4156	84	9	44.969	2525.0	4543	80	9	47.047
C07	4075	4075	4050.0	5224	106	37	54.485	4050.0	4479	94	31	46.485	4050.0	4690	99	37	55.015
C08	4090	4090	4060.0	4004	155	31	53.953	4060.0	4513	196	33	92.125	4060.0	3767	125	27	68.766
C09	5233	5260	**5240.0**	32727	495	305	6933.890	**5240.0**	19023	390	137	2180.630	**5240.0**	14748	335	87	1172.170
C10	4700	4700	4670.0	4057	176	59	87.563	4685.0	5753	241	119	278.281	4685.0	5022	221	125	285.094
C11	4583	4635	4580.0	10966	317	25	345.438	**4585.0**	11582	339	37	569.312	**4585.0**	9918	288	25	569.234
C12	4209	4240	4200.0	18606	267	271	2175.080	4200.0	18858	243	267	2093.520	4200.0	15971	248	189	1587.270
C13	2955	2955	2940.0	4229	65	17	29.578	2940.0	4129	61	15	27.734	2940.0	4171	63	19	39.500
C14	4030	4030	4010.0	3639	64	31	28.922	4010.0	4222	65	45	63.578	4010.0	2725	59	27	34.125
C15	4912	4940	4910.0	28469	163	209	2734.550	**4920.0**	43642	178	561	*3h*	**4913.3**	33902	151	307	*3h*
C16	1475	1475	1470.0	2201	25	1	3.594	1470.0	2065	28	3	6.735	1470.0	1256	25	1	2935.440
C17	3555	3555	3550.0	1635	49	3	3.218	3550.0	1421	48	3	3.485	3550.0	1205	48	3	3.797

Continued on next page

Table VI.1: *Continued from previous page*

Ins	LB	UB	NG=8					NG=16					NG=32				
			Value	Routes	Cuts	Nodes	Time	Value	Routes	Cuts	Nodes	Time	Value	Routes	Cuts	Nodes	Time
C18	5577	5620	5564.5	39731	273	150	*3h*	5567.6	33027	179	85	*3h*	5568.6	34423	197	85	*3h*
C19	3096	3115	**3115.0**	16707	290	147	1134.000	**3115.0**	9861	201	65	1142.050	**3115.0**	5337	120	25	1725.640
C20	2120	2120	2120.0	4452	59	3	11.641	2120.0	5022	53	5	29.703	2120.0	4197	38	5	18.953
C21	3960	3970	3960.0	6976	52	7	45.562	3960.0	6005	50	7	70.171	3960.0	5532	53	5	259.312
C22	2245	2245	2245.0	3137	36	3	5.109	2245.0	2244	36	5	5.484	2245.0	1673	36	1	3.828
C23	4032	4085	**4070.0**	38026	383	241	9081.970	**4067.0**	20228	242	60	*3h*	**4056.1**	8877	179	3	*3h*
C24	3384	3400	**3385.0**	18883	372	51	1042.030	**3385.0**	8924	139	15	144.703	**3385.0**	8856	173	25	255.703
C25	2310	2310	2310.0	1899	29	3	2.797	2310.0	1584	28	1	1.593	2310.0	1343	29	3	2.484
D01	3215	3215	3215.0	8421	54	1	66.453	3215.0	10840	53	1	118.704	3215.0	10856	61	1	135.250
D02	2520	2520	2520.0	3618	25	1	9.250	2520.0	4276	25	1	13.344	2520.0	4744	25	1	17.953
D03	2065	2065	2065.0	6715	31	3	30.704	2065.0	6929	30	3	33.984	2065.0	6413	31	3	39.532
D04	2785	2785	2785.0	12240	54	3	156.766	2785.0	12136	52	3	171.468	2785.0	12839	51	3	233.437
D05	3935	3935	3935.0	5850	49	3	35.188	3935.0	6540	52	3	41.734	3935.0	5870	49	3	43.031
D06	2125	2125	2125.0	5901	30	3	23.484	2125.0	6564	30	3	30.078	2125.0	7326	30	3	45.938
D07	3115	3115	3075.0	14750	137	53	600.516	3090.0	17065	179	57	1515.880	3090.0	14226	104	45	*3h*
D08	2995	3045	**3020.0**	17349	103	39	495.625	**3020.0**	16325	89	33	579.000	**3020.0**	14640	89	27	688.891
D09	4120	4120	4120.0	12842	50	3	172.812	4120.0	15510	50	3	319.016	4120.0	20382	49	3	756.891
D10	3340	3340	3335.0	5304	33	3	34.000	3335.0	5148	33	3	27.875	3335.0	4451	33	3	1132.190
D11	3745	3745	3745.0	18943	78	7	531.953	3745.0	20367	80	7	604.890	3745.0	19543	80	7	752.625
D12	3310	3310	3310.0	9794	49	3	88.734	3310.0	10748	49	3	123.860	3310.0	11227	49	3	151.266
D13	2535	2535	2535.0	4279	37	1	12.968	2535.0	4769	36	1	17.032	2535.0	3461	36	1	12.437
D14	3272	3280	**3280.0**	10278	59	11	125.859	**3280.0**	8839	82	9	120.266	—	—	—	—	—
D15	3990	3990	3990.0	19842	55	3	412.141	3990.0	21235	54	3	550.359	3990.0	24021	54	1	656.812
D16	1060	1060	1060.0	5622	19	1	20.500	1060.0	6662	16	1	131.390	—	—	—	—	—
D17	2620	2620	2620.0	3816	27	3	10.172	2620.0	3464	26	5	13.469	2620.0	2244	26	3	7.907
D18	4165	4165	4165.0	19955	67	3	534.844	4165.0	20805	66	3	680.625	4165.0	21352	67	3	850.312
D19	2393	2400	2385.0	20774	87	13	370.360	—	—	—	—	—	—	—	—	—	—

Continued on next page

Table VI.1: *Continued from previous page*

Ins	LB	UB	NG=8					NG=16					NG=32				
			Value	Routes	Cuts	Nodes	Time	Value	Routes	Cuts	Nodes	Time	Value	Routes	Cuts	Nodes	Time
D20	1870	1870	1870.0	7745	33	3	32.594	1870.0	9467	34	3	76.281	1870.0	8547	34	3	66.500
D21	2985	3050	**3010.0**	48420	116	67	3607.050	**3010.0**	31992	88	49	2229.050	**2995.3**	21954	60	9	*3h*
D22	1865	1865	1865.0	7724	32	3	26.594	1865.0	9213	32	1	43.844	1865.0	5867	32	1	79.453
D23	3114	3130	**3120.0**	33900	116	9	2355.740	**3116.8**	23785	110	3	*3h*	—	—	—	—	—
D24	2676	2710	**2705.0**	61167	218	41	8401.740	**2697.5**	61419	201	21	*3h*	—	—	—	—	—
D25	1815	1815	1815.0	3501	24	1	6.625	1815.0	3841	58	1	10.406	1815.0	3376	24	1	355.859
E01	4885	4910	4870.0	18143	413	439	2719.260	4870.0	15571	321	293	1823.880	4870.0	12827	311	195	1822.480
E02	3990	3990	3975.0	4301	100	25	37.766	3975.0	3950	79	17	31.516	3975.0	3334	65	21	34.922
E03	2015	2015	2015.0	3603	34	3	9.718	2015.0	3372	30	1	5.546	2015.0	2839	30	1	6.312
E04	4155	4155	4145.0	12606	307	31	273.859	4145.0	8560	196	41	1502.480	4145.0	8569	252	29	619.984
E05	4585	4585	4580.0	3110	85	5	15.297	4580.0	3001	75	5	13.828	4585.0	4153	106	17	46.687
E06	2055	2055	2055.0	1668	25	1	2.391	2055.0	1704	25	1	2.125	2055.0	1315	25	1	1.687
E07	4155	4155	4080.0	2961	145	39	42.547	4080.0	2494	130	31	44.953	4080.0	1851	92	13	22.937
E08	4710	4710	4700.0	5454	114	21	56.375	4710.0	5609	129	27	60.016	4710.0	5662	163	23	65.468
E09	5780	5820	5775.0	11761	194	39	492.093	5775.0	9151	173	21	244.031	5780.0	18003	277	213	3092.750
E10	3605	3605	3605.0	1411	58	1	2.078	3605.0	1486	58	1	3.313	3605.0	1098	58	1	1.969
E11	4637	4655	**4645.0**	10111	204	25	305.047	**4645.0**	9216	137	17	223.203	**4650.0**	14520	219	43	1687.860
E12	4180	4180	4165.0	10031	183	197	516.750	4165.0	8475	158	127	324.703	4165.0	8195	151	115	374.657
E13	3345	3345	3330.0	2686	63	5	10.203	3330.0	2519	63	5	11.422	3330.0	2279	63	5	179.797
E14	4115	4115	4105.0	7817	192	179	262.625	4105.0	6092	124	129	190.187	4105.0	5463	139	137	293.500
E15	4189	4205	**4197.6**	47931	177	369	*3h*	**4200.0**	41161	198	273	10450.800	**4197.7**	17654	125	51	*3h*
E16	3755	3775	3775.0	10561	124	53	176.422	3775.0	5599	123	39	118.672	3775.0	6866	115	45	801.266
E17	2740	2740	2740.0	1684	38	3	2.422	2740.0	1565	41	5	4.531	2740.0	1542	39	3	3.297
E18	3825	3835	3825.0	11053	96	13	173.219	**3830.0**	19493	232	43	1215.640	**3830.0**	16008	320	43	3447.670
E19	3222	3235	**3230.0**	10584	144	31	240.453	**3230.0**	8908	108	23	184.312	**3230.0**	7447	121	19	279.844
E20	2802	2825	**2810.0**	6629	110	21	71.672	**2810.0**	6644	126	23	87.172	**2810.0**	4840	73	15	53.235
E21	3728	3730	**3730.0**	8623	67	21	121.235	**3730.0**	8976	79	17	143.515	**3730.0**	7732	79	17	213.984

Continued on next page

Table VI.1: *Continued from previous page*

Ins	LB	UB	NG=8					NG=16					NG=32				
			Value	Routes	Cuts	Nodes	Time	Value	Routes	Cuts	Nodes	Time	Value	Routes	Cuts	Nodes	Time
E22	2470	2470	**2470.0**	3118	63	11	23.922	**2470.0**	2554	61	9	41.359	**2470.0**	2614	56	7	116.750
E23	3686	3710	**3705.0**	33482	564	143	5479.630	**3705.0**	27254	443	91	4330.250	**3696.2**	8536	178	13	*3h*
E24	4001	4020	**4010.0**	15192	199	71	926.172	**4010.0**	11096	155	21	266.094	**4015.0**	13671	191	47	699.579
E25	1615	1615	1615.0	956	20	1	0.578	1615.0	1155	21	1	1.063	1615.0	662	20	1	1.047
F01	4040	4040	4040.0	11375	93	3	210.640	4040.0	13103	92	3	242.062	4040.0	15593	94	3	451.547
F02	3300	3300	3300.0	6067	49	1	26.218	3300.0	5381	49	1	24.281	3300.0	7296	49	1	52.469
F03	1665	1665	1665.0	6982	36	1	21.141	1665.0	7249	36	1	25.219	1665.0	4442	39	1	48.594
F04	3476	3485	**3485.0**	27270	137	27	1471.310	**3485.0**	31918	109	11	*3h*	**3485.0**	17243	93	5	*3h*
F05	3605	3605	3605.0	4154	65	1	17.594	3605.0	5334	65	3	32.110	3605.0	5431	66	3	39.609
F06	1875	1875	1875.0	3149	28	1	6.500	1875.0	3353	28	1	7.875	1875.0	2812	28	1	7.875
F07	3335	3335	3335.0	5511	72	5	39.125	3335.0	4262	59	3	17.468	3335.0	3445	59	3	13.484
F08	3695	3705	**3705.0**	8869	103	21	163.062	**3705.0**	10759	79	27	249.140	**3705.0**	10168	84	19	271.016
F09	4730	4730	4730.0	12822	88	5	230.609	4730.0	17582	88	5	541.109	4730.0	18022	87	5	629.421
F10	2925	2925	2925.0	4130	44	3	15.016	2925.0	4788	44	3	20.218	2925.0	3695	44	3	16.828
F11	3835	3835	3835.0	14947	85	5	263.625	3835.0	19243	206	5	523.250	3835.0	21815	84	5	818.938
F12	3390	3395	3390.0	17590	92	19	436.734	3390.0	18161	94	19	513.000	3390.0	13644	99	13	409.156
F13	2855	2855	2855.0	3355	48	3	10.953	2855.0	3950	48	3	17.110	2855.0	3632	48	3	14.422
F14	3330	3330	3330.0	4443	78	3	23.687	3330.0	5078	50	1	231.485	3330.0	5333	50	1	2019.550
F15	3560	3560	3560.0	24625	74	9	924.421	3560.0	35989	70	13	2394.890	3560.0	29466	70	13	1497.230
F16	2725	2725	2725.0	7484	38	3	30.328	2725.0	7537	37	1	36.406	2725.0	8646	37	1	3384.970
F17	2055	2055	2055.0	2981	31	1	5.875	2055.0	3206	31	1	8.453	2055.0	1497	31	1	3.125
F18	3063	3075	**3065.0**	19142	74	3	329.297	**3065.0**	21413	66	3	2540.670	—	—	—	—	—
F19	2500	2525	2500.0	22195	148	11	595.969	—	—	—	—	—	—	—	—	—	—
F20	2445	2445	2445.0	9775	54	3	74.265	2445.0	10775	54	3	93.703	2445.0	10455	54	3	106.688
F21	2930	2930	2930.0	9647	51	1	65.797	2930.0	11328	51	1	110.875	2930.0	11865	51	1	138.672
F22	2075	2075	2075.0	5902	50	3	21.703	2075.0	7474	56	3	91.063	2075.0	4756	49	1	322.672
F23	2994	3005	**3005.0**	34350	213	10	*3h*	**3005.0**	34409	194	5	5923.860	—	—	—	—	—

Continued on next page

Table VI.1: *Continued from previous page*

Ins	LB	UB	NG=8					NG=16					NG=32				
			Value	Routes	Cuts	Nodes	Time	Value	Routes	Cuts	Nodes	Time	Value	Routes	Cuts	Nodes	Time
F24	3210	3210	3210.0	13914	88	3	215.125	3210.0	18096	102	3	2245.720	3210.0	22045	88	1	9429.840
F25	1390	1390	1390.0	2023	22	1	2.594	1390.0	2092	22	1	3.984	1390.0	736	22	1	3.750
e1-A	3548	3548	3548.0	3116	55	1	26.344	3548.0	2898	55	1	23.312	3548.0	2623	55	1	18.735
e1-B	4498	4498	4496.0	7402	94	131	449.797	4496.0	6400	86	85	344.562	4496.0	5498	87	89	549.750
e1-C	5595	5595	5556.0	2847	70	49	147.890	5556.0	2010	70	19	59.078	5556.0	1627	70	19	62.375
e2-A	5018	5018	5012.0	6531	88	5	160.000	5018.0	5912	94	7	655.797	5018.0	5472	90	7	5886.940
e2-B	6305	6317	6299.0	5529	93	49	340.343	6299.0	4615	89	21	275.172	6301.0	5326	105	53	1349.550
e2-C	8335	8335	8302.0	4882	99	281	810.688	8308.0	5156	123	439	1290.770	8308.0	4569	139	349	1209.700
e3-A	5898	5898	5898.0	7930	81	5	227.359	5898.0	8096	76	5	322.360	5898.0	9244	75	5	1002.020
e3-B	7711	7775	**7715.0**	15787	178	749	7309.340	**7722.0**	15523	210	831	10135.300	**7722.0**	10861	156	289	7833.480
e3-C	10244	10292	10207.0	4787	131	267	1088.140	10214.0	5750	137	407	1914.380	10214.0	5047	111	291	1538.280
e4-A	6408	6444	6393.0	11099	223	23	1060.330	6393.0	12182	163	17	6083.190	—	—	—	—	—
e4-B	8935	8961	8897.0	7552	110	229	1962.030	8900.0	7543	120	203	2289.560	8912.0	9889	218	689	*3h*
e4-C	11493	11550	11490.0	6519	238	901	5085.380	**11495.0**	6754	192	1189	7936.920	**11495.0**	6072	202	723	5176.520
s1-A	5018	5018	5018.0	6904	143	3	176.984	5018.0	10945	143	3	377.640	5018.0	8825	143	3	417.734
s1-B	6388	6388	6388.0	7400	166	35	465.281	6388.0	6510	160	21	369.219	6388.0	7626	182	29	712.125
s1-C	8518	8518	8511.0	5612	183	169	792.062	8511.0	4223	158	91	520.344	8511.0	4598	172	63	409.750
s2-A	9825	9884	9812.7	23671	767	102	*3h*	9816.8	25049	529	79	*3h*	9811.3	20759	300	45	*3h*
s2-B	13017	13100	12986.8	13353	634	357	*3h*	12991.3	12292	591	295	*3h*	12993.0	11357	488	263	*3h*
s2-C	16425	16425	16376.2	8257	397	725	*3h*	16380.0	8608	317	591	*3h*	16382.3	8381	327	601	*3h*
s3-A	10145	10220	**10153.2**	23485	368	93	*3h*	**10152.6**	14615	226	25	*3h*	**10149.7**	10637	224	3	*3h*
s3-B	13648	13682	13633.1	14969	588	275	*3h*	13635.7	11574	512	237	*3h*	13636.6	11146	469	193	*3h*
s3-C	17188	17188	17126.2	9987	614	531	*3h*	17134.1	8422	466	453	*3h*	17136.8	7842	396	473	*3h*
s4-A	12143	12268	12141.8	22904	387	118	*3h*	**12144.5**	18010	398	51	*3h*	**12143.1**	13142	183	15	*3h*
s4-B	16098	16283	16085.9	11926	789	229	*3h*	**16099.1**	10087	685	135	*3h*	**16098.6**	9591	520	125	*3h*
s4-C	20430	20481	20395.5	6455	438	256	*3h*	20407.3	6681	349	307	*3h*	20409.6	6278	339	309	*3h*