6
COMPUTATIONAL RESULTS

Computational tests were perfomed to analyze the performance of the proposed reformulation schemes and solution algorithm. All testes were conducted on a computer with processor Pentium 4, 3.00 GHz and 2 GB of RAM. The models and algorithms were implemented using the modeling language MOSEL and solved by XPRESS 19.00.04.

The first results are those obtained for the set of instances described in Viswanath et al. [66]. These are all small-size problems which served as a “proof of correctness” for the proposed methodology. Since no other work in the literature deals with the problem in its original form (remember that [43] dismisses the probabilistic nature of the problem by assuming that investment on an edge completely eliminates the probability of that edge failing afterwards), several other instances were created in order to assess the performance of the methodology for medium and large-size instances of the problem.

The remainder of this Chapter is organized as follows: Section (6.1) presents the results for the instances provided in [66], Section (6.2) describes how the medium and large-size instances were generated and presents results for the former while Section (6.3) discusses the results for the latter.

6.1
Instances from the literature

All the instances solved in [66] refer to a graph which contains 4 vertices and 5 edges, as depicted in Figure 6-1. There is a total of 28 instances which are detailed in Table 6.1: they differ from each other in the investment and transportation costs associated with each edge (columns *InvCost* and *TranspCost*, respectively), maximum budget (column *Budget*), penalty for not fulfilling the demand associated to a vertex (column *Penalty*) and initial and final survival probabilities (initial survival probability is equal to 70% for all edges in instances 1 through 14 and equal to 60% in instances 15 through 28 and column

SurvProbInv). Vertices *O* and *D* are the origin and destination for a unit commodity that must flow through the network.

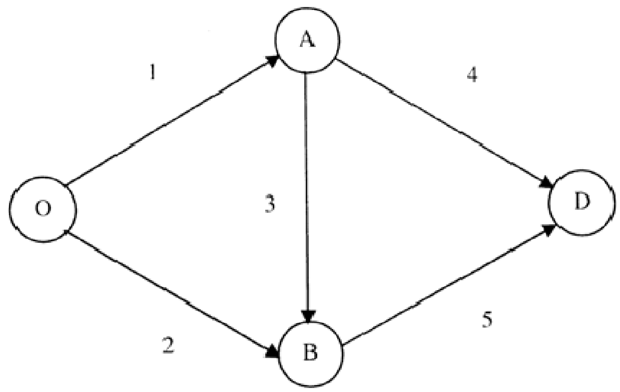

Figure 6-1 – Graph corresponding to the instances solved in Viswanath et al. [66]

For all these instances, there are 32 (2^5) scenarios of network configuration – given by all the possible combinations of the availability of the edges – and the first step of the proposed methodology determines that the minimum cost network flow problem corresponding to each one of these configurations must be solved independently. For this set of instances, total solution time of the network flow problems for all scenarios is minuscule.

Once these optimal values are known, they are used as coefficients in the objective function of the main problem, which is then solved by the algorithm outlined in Chapter 4. All instances were solved to optimality in less than 1.0 second and average solution time was 0.313 second. Details are provided in Table 6.2 where the column *Id* indicates the instance identification, column *OptVal* presents the value of the optimal solution , column *# Iter* indicates the number of iterations of the algorithm until convergence was achieved and column *TotalTime* the time it took for the algorithm to complete.

Table 6-1 – Description of the instances provided in Viswanath et al. [66]

Id	***SurvProbInv***	***InvCost***	***Budget***	***TranspCost***	***Penalty***
1 / 15	{80%, 80%, 80%, 80%, 80%}	{1, 1, 1, 1, 1}	2	{10, 10, 10, 10, 10}	31
2 / 16	{80%, 80%, 80%, 80%, 80%}	{1, 1, 1, 1, 1}	3	{10, 10, 10, 10, 10}	31
3 / 17	{80%, 80%, 80%, 80%, 80%}	{1, 1, 1, 1, 1}	3	{10, 10, 15, 30, 10}	41
4 / 18	{80%, 80%, 90%, 80%, 80%}	{1, 1, 1, 1, 1}	3	{10, 10, 15, 30, 10}	41
5 / 19	{80%, 80%, 80%, 80%, 80%}	{2, 1, 1, 1, 1}	3	{10, 10, 15, 30, 10}	41
6 / 20	{80%, 80%, 80%, 80%, 80%}	{1, 2, 1, 1, 1}	3	{10, 10, 15, 30, 10}	41
7 / 21	{80%, 80%, 80%, 80%, 80%}	{1, 1, 2, 1, 1}	3	{10, 10, 15, 30, 10}	41
8 / 22	{80%, 80%, 80%, 80%, 80%}	{1, 1, 1, 2, 1}	3	{10, 10, 15, 30, 10}	41
9 / 23	{80%, 80%, 80%, 80%, 80%}	{1, 1, 1, 1, 2}	3	{10, 10, 15, 30, 10}	41
10 / 24	{80%, 80%, 80%, 80%, 80%}	{2, 1, 1, 1, 1}	3	{10, 20, 10, 15, 10}	31
11 / 25	{80%, 80%, 80%, 80%, 80%}	{2, 1, 1, 1, 1}	3	{10, 20, 10, 15, 10}	43.9
12 / 26	{80%, 80%, 80%, 80%, 80%}	{2, 1, 1, 1, 1}	3	{10, 20, 10, 15, 10}	57.3
13 / 27	{80%, 80%, 80%, 80%, 80%}	{1, 1, 1, 1, 1}	3	{10, 15, 5, 15, 10}	26
14 / 28	{80%, 80%, 80%, 80%, 80%}	{1, 1, 1, 1, 1}	3	{10, 15, 1, 15, 10}	26

Table 6-2 – Results of the instances provided in Viswanath et al. [66]

Id	***OptVal***	***# Iter***	***TotalTime***
1	21.9961	3	0.137
2	21.7155	2	0.082
3	26.8835	3	0.123
4	26.8494	4	0.225
5	26.9087	3	0.120
6	26.9681	3	0.106
7	26.8835	3	0.136
8	26.8835	3	0.122
9	26.9681	3	0.129
10	26.9601	4	0.257
11	29.0251	3	0.144
12	31.0963	3	0.297
13	25.1315	5	1.000
14	23.0995	3	0.359
15	22.5114	3	0.302
16	22.0285	2	0.187
17	26.9725	3	0.359
18	26.9638	3	0.531
19	27.0157	3	0.359
20	27.1194	3	0.375
21	26.9725	3	0.360
22	26.9725	3	0.421
23	27.1194	3	0.359
24	27.0074	3	0.422
25	28.8943	3	0.421
26	32.0447	3	0.375
27	25.1565	3	0.625
28	23.1405	2	0.235

6.2
Medium-size instances

Given the lack of additional instances of the problem available in the literature, we developed an instance generator which was then used to test the proposed methodology.

The instances were created by randomly selecting the location of a given number of vertices within a region defined by minimum and maximum values for the x and y coordinates. Next, a predefined number of edges connecting the vertices was created (the resulting graph was checked for connectedness in order to avoid trivial and meaningless solutions) and the Euclidean distance between the corresponding vertices was assigned as the transportation cost of each edge. Pre- and post-investment survival probabilities were assigned to each edge and, for the large instances presented in Section 6.3, scenarios of network configuration were generated based on the initial survival probability of each edge.

Next, in Tables 6.3 and 6.4, we present the results for a total of 30 instances which were all solved by the algorithm designed in Chapter 4 (ALG2) with full scenario enumeration and tolerance level set to no more than 1%. The table provides the following information: column *Id* identifies the instance, columns *# Vertices* and *# Edges* indicates, respectively, the number of vertices and edges of the graph, column *# Scen* provides the number of scenarios of network configuration used in each problem; column *UB* reports the value of the best solution found while column *LB* indicates the value of the solution to the last approximated problem (i.e., the one which is solved by considering the set of cuts that approximate the exponential function), column *% Gap* presents the percentage gap between the upper and lower bounds and column *ErrTol* contains the maximum acceptable error, which is the stopping criterium for the algorithm; column *# Iter* indicates the number of iterations of the algorithm needed to reach the final solution, and column *MainTime* report the total time for the convergence of the algorithm (the time needed for the solution of the independent scenario-specific network flow problems is not reported but they are usually orders of magnitude smaller than the time it takes for the algorithm to converge which thus represents the bottleneck of the methodology).

Table 6-3 – Results for the medium-size instances

Id	*# Vertices*	*# Edges*	*# Scen*	*UB*	*LB*	*% Gap*	*ErrTol*	*# Iter*	*MainTime*
v5e6A	5	6	64	219.423	219.423	0.00%	0.1%	5	0.437
v5e6B	5	6	64	50.6467	50.6467	0.00%	0.1%	3	0.140
v5e6C	5	6	64	143.923	143.923	0.00%	0.1%	4	0.281
v5e6D	5	6	64	107.918	107.918	0.00%	0.1%	4	0.250
v5e6E	5	6	64	277.904	277.904	0.00%	0.1%	4	0.281
v6e8A	6	8	256	361.265	361.265	0.00%	0.1%	6	1.796
v6e8B	6	8	256	45.9159	45.9159	0.00%	0.1%	5	1.328
v6e8C	6	8	256	350.268	350.268	0.00%	0.1%	5	1.594
v6e8D	6	8	256	110.915	110.915	0.00%	0.1%	5	1.062
v6e8E	6	8	256	65.4201	65.4201	0.00%	0.1%	3	0.328
v7e10A	7	10	1024	122.857	122.851	0.0049%	0.1%	4	3.735
v7e10B	7	10	1024	201.934	201.926	0.0040%	0.1%	5	4.468
v7e10C	7	10	1024	104.863	104.857	0.0057%	0.1%	4	3.063
v7e10D	7	10	1024	158.868	158.861	0.0044%	0.1%	5	4.063
v7e10E	7	10	1024	75.5659	75.5619	0.0053%	0.1%	5	5.704

Table 6-4 – Results for the medium-size instances

Id	*# Vertices*	*# Edges*	*# Scen*	*UB*	*LB*	*% Gap*	*ErrTol*	*# Iter*	*MainTime*
v7e11A	7	11	2048	306.692	306.672	0.0065%	0.1%	8	79.143
v7e11B	7	11	2048	252.026	251.985	0.0163%	0.1%	6	39.344
v7e11C	7	11	2048	66.7778	66.7561	0.0325%	0.1%	5	15.593
v7e11D	7	11	2048	312.985	312.959	0.0083%	0.1%	6	38.875
v7e11E	7	11	2048	58.0173	57.9977	0.0338%	0.1%	5	27.859
v8e12A	8	12	4096	31.9086	31.5927	0.99%	1%	11	405.395
v8e12B	8	12	4096	141.750	141.515	0.1658%	1%	6	242.878
v8e12C	8	12	4096	97.1507	97.0042	0.1508%	1%	4	40.219
v8e12D	8	12	4096	49.6668	49.4192	0.4985%	1%	7	286.253
v8e12E	8	12	4096	155.492	155.381	0.0714%	1%	4	71.297
v8e12b6A	8	12	4096	40.8165	40.6271	0.4640%	1%	6	247.645
v8e12b6B	8	12	4096	28.5302	28.2736	0.8994%	1%	8	831.795
v8e12b6C	8	12	4096	22.3931	22.1859	0.9253%	1%	7	449.365
v8e12b6D	8	12	4096	66.6392	66.3808	0.3878%	1%	8	992.796
v8e12b6E	8	12	4096	105.355	104.637	0.6815%	1%	5	113.814

6.3
Large-size instances

The set of instances in Section 6.2 involved graphs with a maximum of twelve edges and 4096 possible scenarios of network configuration. The total time required to solve these problems clearly shows how the computational effort increased very rapidly with respect to the number of edges – just as an illustration of this fact, the average time needed to solve the instances with 11 edges was 40.2 seconds, while the average time consumed by the algorithm in solving the instances with 12 edges was 368.1 seconds.

A critical example is provided by an instance of the problem with 10 vertices and 15 edges (and, consequently, 32768 possible scenarios of network configuration) which was solved by full scenario enumeration. Figure 6-2 below presents the performance of the algorithm – data points represent the upper and lower bounds obtained at each iteration:

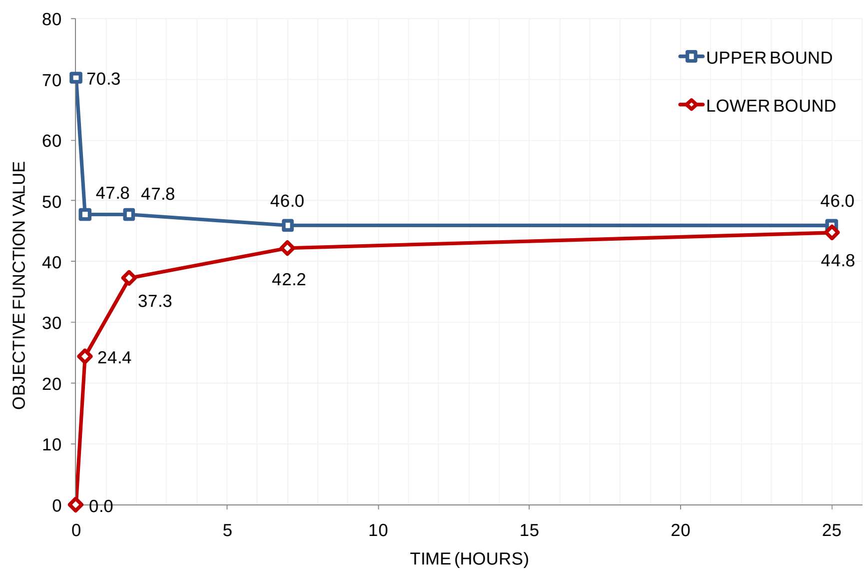

Figure 6-2 – Algorithm perfomance on an 15-edge instance with full scenario enumeration

While the previous instances converged to solutions with gaps not larger than 1% after no more than 17 minutes, in the case of the 15-edge instance it took a total of 25 hours for the algorithm to narrow the gap down to 2.57%. This clearly leads to the conclusion that full scenario enumeration is currently not a viable option when one tries to solve large scale problems and a sample-based version of the problem – such as that suggested in Chapter 5 – becomes a necessity.

In Table 6.5, we present results for 18 instances with the number of edges ranging from 15 to 40, also constructed according to the description given in the previous Section. All instances were solved to a maximum gap of 0.87%. Compared to Table 6.4, there is an additional column *# TotScen* where the number of possible scenarios of network configuration is reported – column *# Scen* indicates the number of scenarios actually used when solving the problem.

The instances with 15 edges (*v10e15_1*, *v10e15_2* and *v10e15_3*) all refer to the same graph of the example for which the convergence of the algorithm was shown in Figure 6-2. Each one of them was solved using a different set of 500 scenarios (out of the 32768 possible network configurations), sampled according to the initial probability distribution of the edges' availabilities. It is interesting to observe that even though the number of scenarios used in these instances is significantly smaller than the total number of possible scenarios, the solutions found for these problems in under 60 seconds have an objective function value which is close to that found after 25 hours in the case of full scenario enumeration.

A significant increase in computational times was observed when solving the instances with 20 edges (*v10e20_1* through *v10e20_5*) and 500 scenarios. All instances refer to the same graph and were solved using different sets of scenarios to a maximum gap of 0.868%, including one instance which was solved to optimality. The difference between the minimum and maximum optimal values obtained for all five problems was 5.49%, which seems like a reasonable compromise considering that the number of scenarios actually used to solve the problems represents a very small fraction (namely, 0.048%) of all possible network configurations.

As the number of binary variables increase so does the computational effort required to solve the problems at each iteration, leading to a compromise between network size (which dictates the number of binary variables) and number of sampled scenarios. This was done when solving the instances with 25, 30 and 40 edges: the number of scenarios utilized was reduced to 300 in the former and 200 in the last two cases.

Table 6-5 – Results for the large-size instances

Id	*# Vertices*	*# Edges*	*# Scen*	*# TotScen*	*UB*	*LB*	*% Gap*	*ErrTol*	*# Iter*	*MainTime*
v10e15_1	10	15	500	3.28E+4	44.5735	44.4068	0.374%	1%	6	41.032
v10e15_2	10	15	500	3.28E+4	44.8405	44.8405	0%	1%	6	27.562
v10e15_3	10	15	500	3.28E+4	47.0452	46.9256	0.254%	1%	6	27.578
v10e20_1	10	20	500	1.05E+6	81.8326	81.7279	0.128%	1%	8	1169.703
v10e20_2	10	20	500	1.05E+6	81.4761	80.7692	0.868%	1%	9	2725.251
v10e20_3	10	20	500	1.05E+6	81.6981	81.3323	0.448%	1%	10	1713.125
v10e20_4	10	20	500	1.05E+6	78.4703	78.1662	0.388%	1%	10	3164.391
v10e20_5	10	20	500	1.05E+6	77.3371	77.3371	0%	1%	10	4028.000
v12e25_A	12	25	300	3.36E+7	75.5378	74.9683	0.754%	1%	8	2528.266
v12e25_B	12	25	300	3.36E+7	52.4450	52.1022	0.654%	1%	12	3133.67
v12e25_C	12	25	300	3.36E+7	70.3214	70.1937	0.182%	1%	11	1882.297
v12e25_D	12	25	300	3.36E+7	43.1088	42.931	0.412%	1%	8	810.234
v13e30_1	13	30	200	1.07E+9	32.4429	32.3264	0.359%	1%	9	516.454
v13e30_2	13	30	200	1.07E+9	38.8679	38.6852	0.470%	1%	11	6332.030
v13e30_3	13	30	200	1.07E+9	32.4177	32.4131	0.014%	1%	7	1086.485
v13e30_4	13	30	200	1.07E+9	33.1784	32.9004	0.838%	1%	9	1095.703
v13e30_5	13	30	200	1.07E+9	34.4604	34.1707	0.841%	1%	9	3457.325
v16e40_1	16	40	200	1.10E+12	19.5682	19.4753	0.475%	1%	7	4367.515