Math. Prog. Comp. (2012) 4:239–273
DOI 10.1007/s12532-012-0039-y

FULL LENGTH PAPER

Cutting plane versus compact formulations for uncertain (integer) linear programs

Matteo Fischetti · Michele Monaci

Received: 25 October 2010 / Accepted: 15 March 2012 / Published online: 20 April 2012

Abstract Robustness is about reducing the feasible set of a given nominal optimization problem by cutting "risky" solutions away. To this end, the most popular approach in the literature is to extend the nominal model with a polynomial number of additional variables and constraints, so as to obtain its robust counterpart. Robustness can also be enforced by adding a possibly exponential family of cutting planes, which typically leads to an exponential formulation where cuts have to be generated at run time. Both approaches have pros and cons, and it is not clear which is the best one when approaching a specific problem. In this paper we computationally compare the two options on some prototype problems with different characteristics. We first address robust optimization à la Bertsimas and Sim for linear programs, and show through computational experiments that a considerable speedup (up to 2 orders of magnitude) can be achieved by exploiting a dynamic cut generation scheme. For integer linear problems, instead, the compact formulation exhibits a typically better performance. We then move to a probabilistic setting and introduce the *uncertain set covering problem* where each column has a certain probability of disappearing, and each row has to be covered with high probability. A related uncertain graph connectivity problem is also investigated, where edges have a certain probability of failure. For both problems, compact ILP models and cutting plane solution schemes are presented and compared through extensive computational tests. The outcome is that a compact ILP formulation (if available) can be preferable because it allows for a better use of the rich arsenal of preprocessing/cut generation tools available in modern ILP solvers. For the cases where such a compact ILP formulation is not available, as in the uncertain connectivity

M. Fischetti (✉) · M. Monaci
DEI, Università di Padova, Padua, Italy
e-mail: matteo.fischetti@unipd.it

M. Monaci
e-mail: michele.monaci@unipd.it

problem, we propose a restart solution strategy and computationally show its practical effectiveness.

Keywords Optimization under uncertainty · Cutting planes · Uncertain set covering · Uncertain connectivity problems

Mathematics Subject Classification 90C5 linear programming · 90C10 integer programming · 90C15 stochastic programming

1 Introduction

One of the main issues when solving real-world optimization problems is the determination of *robust* solutions, i.e., solutions that are stable with respect to certain variations of the input parameters. An increasing amount of research has been devoted to this subject in the last years.

Two main approaches have been proposed for dealing with uncertain data: stochastic programming and robust optimization. Stochastic programming introduces additional variables and penalizes feasible solutions that are most likely to become infeasible due to uncertainty. Hence, it requires some knowledge of how uncertainty will act (which is not always available in practice) and often leads to huge models that can be extremely hard to solve in practice. Robust optimization associates uncertainty with hard constraints restricting the solution space, i.e., one is required to find a solution that is still feasible for worst-case parameters chosen within a certain uncertainty domain. This is a simple way to model uncertainty, but it can lead to overconservative solutions that are quite bad in terms of cost (actually, a feasible solution may not exist at all). A main result in robust optimization is the work by Bertsimas and Sim (BS) [8], where a compact way to model the robust counterpart of a given nominal model is proposed. The approach requires the introduction of a polynomial number of new variables and constraints into the nominal model, hence it does not increase the theoretical complexity of the problem to be solved with respect to the nominal one (e.g., the robust counterpart of a linear program remains a linear program). A heuristic version named "light robustness" has been proposed by the authors in [16].

The starting point of our research was the observation that BS robustness can alternatively be enforced by working directly on the space of the original (nominal) variables, without the need of artificial variables, at the expenses of the addition of an exponential number of linear constraints that reduce the feasibility region. As explained in more details in the sequel, the separation problem for these *robustness cuts* can be carried out very efficiently, as it amounts to the solution of the LP relaxation of a simple cardinality knapsack problem for each uncertain row of the nominal problem. As a result, an effective cutting-plane approach for robustness is conceivable, that generates robustness cuts on the fly. This approach is not new, but as far as we know no computational experiments have been performed to compare the *practical* performance of the BS compact formulation and of its cutting plane counterpart. In this paper we investigate this issue, and show through extensive computational tests that the cutting plane approach can be significantly faster—up to 2 orders of magnitude, for certain

LP instances. For integer linear problems, instead, the compact formulation exhibits a typically better performance.

Besides practical considerations, the cutting plane approach has some important features that can make it the method of choice in important applications. Indeed, cutting planes are a valid method to handle problems whose nominal formulation is itself noncompact as, e.g., in routing problems that involve an exponential number of connectivity constraints. More importantly, the separation procedure for robustness cuts can take into account uncertainty domains that are more complex than in the BS approach. A notable case arises when the uncertainty domain involves yes-no decisions that cannot be modeled by continuous variables (thus making the BS approach—that relies on LP duality—unapplicable), but can be described by a knapsack constraint. In this situation, separation can still be performed effectively in practice, though its theoretical complexity becomes exponential. As an example of this setting, in this paper we introduce and study a set-covering problem with chance constraints, where the uncertain parameters follow a Bernoulli distribution. In the resulting *uncertain set covering problem* (USCP), each column has a certain probability of disappearing (a yes-no event), and each row has to be covered with a high probability. This framework finds applications also in problems that can be reformulated through set covering constraints as, e.g., in the uncertain counterpart of connectivity problems in graphs modeled through cut conditions.

As already mentioned, our order of business is to compare the practical behavior of cutting plane and compact formulations for dealing with uncertainty. Even if the problems considered for our comparison are quite specialized, our computational analysis can be of help to researchers and practitioners looking for the most effective way to deal with uncertainty in their own setting.

The paper is organized as follows. In Sect. 2 we concentrate on robust optimization for (integer) linear programs and introduce a cutting plane approach that implements the projection of the BS model onto the space of the original variables, showing through computational tests the potential of this method. Section 3 introduces our USCP model, for which two alternative formulations are presented and compared through extensive computational tests. In Sect. 4 we address a probabilistic graph connectivity problem having a set covering formulation that allows for the application of our USCP models. We establish the complexity of the problem and propose a restart solution strategy whose performance is analyzed computationally. Finally, Sect. 5 summarizes our findings and draws some conclusions.

2 Bertsimas–Sim robustness

We first concentrate on the robust counterpart of a generic "nominal" Linear Program (LP) of the form

$$\min \sum_{j \in N} c_j\, x_j \tag{1}$$

$$\text{s.t.} \sum_{j \in N} a_{ij}\, x_j \le b_i, \quad i \in M, \tag{2}$$

$$x_j \ge 0, \quad j \in N, \tag{3}$$

where $N = \{1, \cdots, n\}$ and $M = \{1, \cdots, m\}$ denote the variable and constraint index sets, respectively. Our results can be applied in a straightforward way to Mixed-Integer (or Pure) Linear Programs as well—as a matter of fact, enumerative methods reduce the solution of these problems to the solution of a sequence of LPs.

The first attempt to handle data uncertainty through mathematical models was performed by Soyster [36], who considered uncertain problems of the form

$$\min \left\{ \sum_{j \in N} c_j\, x_j : \sum_{j \in N} A_j\, x_j \leq b, \quad \forall A_j \in \mathcal{K}_j, \;\; j \in N \right\}$$

where $\mathcal{K}_j$ are convex sets associated with "column-wise" uncertainty. This approach tends to lead to overconservative models, thus to poor solutions in term of optimality. Later, Ben-Tal and Nemirovski [3–5] defined less conservative models by considering ellipsoidal uncertainties. Moreover, Ben-Tal and Nemirovski [3] show that the robust counterpart of an uncertain LP is equivalent to an explicit computationally tractable problem, provided that the uncertainty is itself "tractable". On the contrary, when the problem to be considered is an ILP, these nonlinear (convex) models become computationally hard problems.

Bertsimas and Sim [7,8] considered a different concept of robustness that can be outlined as follows. It is assumed that each coefficient in the constraint matrix A can take any value $\widetilde{a}_{ij} \in [a_{ij} - \hat{a}_{ij}, a_{ij} + \hat{a}_{ij}]$ and the deviation of such value with respect to the nominal one is independent of the changes of the remaining coefficients. As it is unlikely that all coefficients take their worst value, a solution is considered *robust* if it remains feasible when at most Γ_i coefficients (chosen in any way) in row i take their worst value. The underlying assumption here is that, when more than Γ_i coefficients change, their deviation will tend to compensate one each other, the net effect being comparable with the worst-case deviation of no more than Γ_i coefficients.

Note that $\Gamma_i = 0$ means that robustness is not taken into account and the nominal constraint is considered, whereas $\Gamma_i = |N|$ means that each coefficient in row i can take its worst value, and corresponds to the conservative method by Soyster [36].

According to BS, given the nominal LP (1)–(3) one defines another LP (in an extended space) whose optimal solution remains feasible for every change of, at most, Γ_i coefficients (up to their worst value) in each row $i \in M$. To this end, the ith constraint of the nominal problem is first replaced by

$$\sum_{j \in N} a_{ij}\, x_j + \beta(x, \Gamma_i) \leq b_i \tag{4}$$

where term

$$\beta(x, \Gamma_i) = \max_{S \subseteq N : |S| \leq \Gamma_i} \sum_{j \in S} \hat{a}_{ij}\, x_j \tag{5}$$

indicates the level of protection of the solution found with respect to the uncertainty associated with row i (recall that we assume $x \geq 0$). By using LP duality, the robust counterpart of the nominal problem (1)–(3) then becomes

$$\min \sum_{j \in N} c_j \, x_j \tag{6}$$

$$\text{s.t.} \quad \sum_{j \in N} a_{ij} \, x_j + \Gamma_i \, z_i + \sum_{j \in N} p_{ij} \le b_i, \quad i \in M, \tag{7}$$

$$-\hat{a}_{ij} \, x_{ij} + z_i + p_{ij} \ge 0, \quad i \in M, \ j \in N, \tag{8}$$

$$z_i \ge 0, \quad i \in M, \tag{9}$$

$$p_{ij} \ge 0, \quad i \in M, \ j \in N, \tag{10}$$

$$x_j \ge 0, \quad j \in N. \tag{11}$$

Note that, for the sake of simplicity, we are assuming that all Γ_i's are integer, although this is not required in the original BS approach. If this is not the case, the expression of each term $\beta(x, \Gamma_i)$ is very similar to (5), and the robust counterpart of the nominal problem is updated accordingly. (In the same way, the cutting plane approach described in the next section can easily handle noninteger Γ_i's.)

2.1 Robust optimization through cutting planes

The robust BS formulation (6)–(11) given in the previous section has the very nice property of being *compact*, in the sense that it involves a number of variables and constraints that is polynomial in the input size. Its projection onto the space of the original x variables requires however an exponential number of cuts, and can be obtained along the following lines.

Given the ith constraint of the nominal problem, the robust constraint (4) can be expressed through the following *robustness cuts*:

$$\sum_{j \in N} a_{ij} \, x_j + \sum_{j \in S} \hat{a}_{ij} \, x_j \le b_i, \quad \forall S \subseteq N : |S| \le \Gamma_i. \tag{12}$$

The separation problem for robustness cuts can be stated as follows: given a solution x^* satisfying (1)–(3), find a set $S \subseteq N$ such that: (a) $|S| \le \Gamma_i$, and (b) $\sum_{j \in S} \hat{a}_{ij} \, x_j^*$ is a maximum. The former condition imposes that at most Γ_i coefficients change with respect to their nominal values, while the latter looks for the constraint (12) associated with row i that is most violated by the current solution x^*.

For each row i, the above separation problem can be solved easily by associating each variable x_j with a *profit* $\hat{a}_{ij} \, x_j^*$ and by finding a maximum-profit subset S with at most Γ_i elements. If the sum of the profits of the selected items exceeds $b_i - \sum_{j \in N} a_{ij} \, x_j^*$, a violated constraint is found; otherwise all constraints (12) associated with row i are satisfied by x^*. The separation problem can therefore be solved by just selecting the (at most) Γ_i variables with largest positive profit, and requires $O(n)$ time by using a partial-sorting technique for determining the Γ_ith largest entry in the profit array [15]. It then follows that the robust BS model can be solved in polynomial time also by cutting planes. Obviously, the approach leads to a branch-and-cut algorithm when (mixed) integer LPs (as opposed to just LPs) are addressed. Note that each violated constraint is possibly strengthened by adding to S some indexes j with $x_j^* = 0$, chosen in a greedy way, while preserving condition $|S| \leq \Gamma_i$.

A BS generalization was recently proposed in Chen et al. [12], where uncertainty sets that capture the asymmetry of the underlying random variables are considered, and a new approach to stochastic linear optimization problems with chance constraints modeled through Second-Order Cone Programming (SOCP) is proposed. The application of cutting planes in the resulting SOCP (as opposed to LP) framework appears as a challenging research subject, that however is out of the scope of the present paper dealing with (integer) linear programs, and is left to future investigation.

2.2 Computational experiments

In this section we computationally compare the performance of the original BS approach and of our cutting plane algorithm.

Our algorithm was coded in C language and embedded into the commercial solver `IBM ILOG Cplex` version 11.0 [26], by using its default parameter setting. At each iteration, the most violated robustness cut associated with each nominal constraint is added to the current problem. The tolerance used in our implementation for checking robustness cut violation is 10^{-6}, while the default setting is used for all `IBM ILOG Cplex` parameters. All codes were executed on a PC Intel Q6600 CPU@2.40 GHz with 4GB RAM.

We considered two different sets of instances, to address both LP and (mixed) ILP models, respectively.

As to the first set of instances, as in Ben-Tal and Nemirovski [4] we considered all NETLIB instances, available at http://www.netlib.org/lp/data/. In our model we assume variable nonnegativity, thus we transformed each instance involving negative variables into an equivalent problem with nonnegative variables only. As this operation requires the definition of some new variables, the corresponding instances are marked with an asterisk in the table. Among all instances, we considered only those that have at least 500 variables and 500 constraints and involve, at least, one inequality constraint. As customary, the coefficients arising in equality constraints, if any, are assumed not to be affected by uncertainty. Finally, we disregarded instance `80bau3b` because it involves very small coefficients that lead to numerical troubles when uncertainty is taken into account.

Uncertainty was modeled by allowing each coefficient appearing in an inequality constraint to differ by at most 1 % from its nominal value, as in [8]. Moreover, we considered $\Gamma_i = \Gamma$ for each row i, with $\Gamma \in \{1, 10, 50\}$.

The first 5 columns in Table 1 report, for the 34 instances in our test bed, the following information associated with each nominal instance: name of the instance, number of variables (n) and constraints (m), optimal solution value (z), and computing time, in CPU seconds, to solve it (T). The next four columns report information about uncertainty: number of rows involving some uncertain coefficients (m_u), total number of uncertain coefficients (p_u), number of variables (n') and constraints (m') required by the compact formulation —all these figures do not depend on the considered value of Γ.

Table 1 Results on robust LP instances from the NETLIB (`inf.` indicates that no feasible robust solution exists)

	Instance								Γ = 1				Γ = 10				Γ = 50			
	n	m	z	T	m_u	p_u	n'	m'	%Δz	t_{BS}	t_{CP}	#cuts	%Δz	t_{BS}	t_{CP}	#cuts	%Δz	t_{BS}	t_{CP}	#cuts
25fv47	1,571	821	5,501.846	0.18	305	4,492	6,368	5,313	1.457	1.19	0.23	149	2.541	1.41	0.25	190	2.548	1.83	0.21	131
bnl1	1,175	643	1,977.630	0.04	401	2,553	4,129	3,196	inf.	0.22	0.06	238	inf.	0.38	0.05	241	inf.	0.56	0.05	241
bnl2	3,489	2,324	1,811.237	0.06	953	7,113	11,555	9,437	0.790	0.51	0.32	631	1.840	0.63	0.74	882	1.847	0.75	0.21	406
cycle*	2,864	1,903	−5.226	0.01	510	8,381	11,755	10,284	0.000	0.24	0.03	6	0.000	0.65	0.03	9	0.000	0.32	0.03	9
czprob	3,523	929	2,185,196.699	0.03	37	3,645	7,205	4,574	0.122	0.08	0.03	22	0.375	0.09	0.25	151	0.640	0.15	24.66	1,746
d2q06c	5,167	2,171	122,784.211	1.12	664	11,595	17,426	13,766	inf.	5.74	1.38	284	inf.	12.03	1.93	305	inf.	9.53	1.86	307
degen3	1,818	1,503	−987.294	0.21	786	21,088	23,692	22,591	inf.	62.66	0.67	494	inf.	32.59	0.85	494	inf.	60.74	0.82	498
fffff800	854	524	555,679.565	0.01	174	1,452	2,480	1,976	inf.	0.03	0.00	47	inf.	0.04	0.01	47	inf.	0.03	0.01	47
ganges	1,681	1,309	−109,585.736	0.01	25	300	2,006	1,609	0.053	0.01	0.01	25	0.430	0.01	0.02	31	0.474	0.01	0.01	24
gfrd-pnc	1,092	616	6,902,236.000	0.01	68	195	1,355	811	0.059	0.02	0.00	66	0.065	0.00	0.01	62	0.065	0.01	0.01	62
greenbea*	5,405	2,392	−72,555,248.130	0.47	193	8,279	13,877	10,671	3.041	0.47	0.64	130	11.284	1.14	2.29	751	inf.	2.33	1.19	112
greenbeb*	5,409	2,392	−4,302,260.261	0.59	193	8,279	13,881	10,671	0.057	1.10	0.80	163	8.208	5.61	7.90	1,418	inf.	4.71	2.06	136
maros	1,443	846	−58,063.744	0.05	522	6,005	7,970	6,851	5.760	0.48	0.10	291	12.114	0.65	0.16	332	12.120	0.73	0.10	211
nesm*	3,051	662	14,076,036.488	0.18	94	188	3,333	850	0.451	0.18	0.19	37	0.875	0.21	0.19	37	0.875	0.31	0.19	37
perold*	1,464	625	−9,380.755	0.11	130	2,286	3,880	2,911	inf.	0.70	0.32	113	inf.	1.10	0.37	103	inf.	0.66	0.37	103
pilot87*	4,892	2,030	301.710	7.73	1,797	69,515	76,204	71,545	inf.	0.15	8.72	1,636	inf.	0.17	8.26	1,644	inf.	0.17	8.12	1,644
pilotja*	2,076	940	−6,113.136	0.21	279	6,615	8,970	7,555	2.344	0.96	0.37	143	4.815	2.43	0.43	177	4.877	1.74	0.44	183
pilot	3,652	1,441	−557.490	1.17	1,208	39,478	44,338	40,919	inf.	90.26	4.45	1,092	inf.	6.32	1.76	1,095	inf.	18.99	1.48	1,095
pilotnov	2,172	975	−4,497.276	0.12	274	2,832	5,278	3,807	4.402	0.40	0.41	147	8.510	0.41	0.79	307	8.510	0.44	0.33	123
pilotwe*	2,869	722	−2,720,107.539	0.26	139	1,494	4,502	2,216	3.193	0.63	0.33	116	6.108	0.90	0.34	124	6.109	0.66	0.34	114
scfxm2	914	660	36,660.262	0.02	286	2,244	3,444	2,904	0.989	0.07	0.03	156	2.113	0.14	0.04	217	2.114	0.18	0.03	142
scfxm3	1,371	990	54,901.255	0.04	429	3,366	5,166	4,356	0.977	0.13	0.06	238	2.141	0.25	0.09	323	2.142	0.29	0.06	217
sctap2	1,880	1,090	1,724.807	0.01	620	5,304	7,804	6,394	1.533	0.05	0.17	403	2.814	0.10	0.82	983	2.844	0.08	0.03	126

Table 1 continued

	Instance								$\Gamma = 1$				$\Gamma = 10$				$\Gamma = 50$			
	n	m	z	T	m_u	p_u	n'	m'	$\%\Delta z$	t_{BS}	t_{CP}	#cuts	$\%\Delta z$	t_{BS}	t_{CP}	#cuts	$\%\Delta z$	t_{BS}	t_{CP}	#cuts
sctap3	2,480	1,480	1,424.000	0.01	860	7,014	10,354	8,494	1.602	0.09	0.37	467	2.995	0.12	1.64	1,197	3.040	0.12	0.07	150
seba	1,035	515	15,711.600	0.00	1	7	1,043	522	0.000	0.00	0.01	0	0.000	0.00	0.00	0	0.000	0.00	0.00	0
shell	1,775	536	1,208,825,346.000	0.00	2	6	1,783	542	0.000	0.01	0.00	0	0.000	0.01	0.00	0	0.000	0.01	0.00	0
ship08l	4,283	778	1,909,055.211	0.01	80	4,391	8,754	5,169	0.030	0.04	0.01	18	0.116	0.05	0.02	22	0.124	0.06	0.02	28
ship08s	2,387	778	1,920,098.211	0.01	80	2,495	4,962	3,273	0.032	0.02	0.01	31	0.129	0.03	0.01	35	0.140	0.03	0.01	35
ship12l	5,427	1,151	1,470,187.919	0.02	106	5,535	11,068	6,686	0.060	0.05	0.02	45	0.346	0.06	0.05	65	0.353	0.07	0.04	67
ship12s	2,763	1,151	1,489,236.134	0.01	106	2,871	5,740	4,022	0.062	0.03	0.02	57	0.386	0.04	0.02	80	0.390	0.04	0.02	63
sierra	2,036	1,227	15,394,362.184	0.01	699	3,329	6,064	4,556	0.022	0.03	0.03	56	0.024	0.04	0.02	55	0.024	0.03	0.02	55
stocfor2	2,031	2,157	−39,024.409	0.03	1,014	3,414	6,459	5,571	0.759	0.24	0.13	460	1.522	0.24	0.12	443	1.522	0.29	0.12	443
stocfor3	15,695	16,675	−39,976.784	0.68	7,846	26,472	50,013	43,147	0.733	3.43	7.90	3,800	1.482	4.03	6.89	3,786	1.482	7.29	6.89	3,786
woodw	8,405	1,098	1.304	0.06	13	1,473	9,891	2,571	0.447	0.07	0.07	5	1.280	0.09	0.06	3	1.280	0.08	0.06	3
Arith. mean (all)				0.40						5.01	0.82	340.18		2.12	1.07	459.09		3.33	1.47	363.06
Arith. mean (infeas. only)										22.82	2.23	466.23		7.52	1.89	677.56		10.86	1.77	464.78

The remaining columns of Table 1 give, for each value of Γ, the percentage increase of the robust objective value z_R with respect to the nominal one (namely, $\%\Delta z = 100 * (z_R - z)/|z|$), the computing time t_{BS} required for solving the BS model (6)–(11), the computing time t_{CP} required by the cutting plane method, along with the number $\#cuts$ of robustness cuts added. Note that we let `IBM ILOG Cplex` choose the most appropriate algorithm for solving both model (6)–(11) and the first LP within our cutting plane scheme (i.e., the nominal problem), whereas all LPs after separation are reoptimized by using the dual simplex. Finally, the last two lines of the table give the average computing time (arithmetic mean) of each method over all the instances in our testbed and over all the "infeasible" instances for which a robust solution does not exist, respectively.

Results of Table 1 show that, for most instances, the computing time required by our cutting plane approach is considerably smaller than that required to solve the compact BS formulation, with an average speedup from 2 to 6, depending on the value of Γ. A larger speedup, up to one order of magnitude, is achieved in some cases—in particular, when the addition of a limited number of robustness cuts is enough to prove that no robust solution exists. For these infeasible cases, our iterative cutting plane approach also has an interesting property, namely, it provides a "last robust solution before infeasibility" that can be useful in practice. Conversely, in some rare cases (e.g., `pilot87*`), infeasibility of the robust problem can be proved only after several cutting plane rounds, whereas the compact formulation detects this situation very easily—with a computing time that is even smaller than that required to solve the nominal problem—due to a very effective preprocessing of the input instance.

We also note that, in most cases, the cutting plane approach is more effective for $\Gamma = 1$ than for $\Gamma = 10$, due to the smaller number of robustness cuts required. A small number of robustness cuts is often required also when $\Gamma = 50$, as for sparse constraints our strengthening procedure tends to produce a unique cut where all uncertain coefficients take their worst value. This is no longer the case when a large number of uncertain coefficients exist in some rows as, e.g., in `czprob`, where a very large number of robustness cuts is generated.

Our second set of instances is made by (mixed) integer linear programs from the MIPLIB 3.0 library, available at http://miplib.zib.de/miplib3/miplib.html. Instances in this test set are nowadays pretty easy to solve in their nominal version, but their robust counterpart turns out to be quite hard in some cases. In particular, we addressed all the instances that (a) were solved to proven optimality by `IBM ILOG Cplex` within a 180-s time limit; (b) contain at least one inequality; (c) involve only nonnegative variables; and (d) contain at most 10,000 variables and constraints. The robust counterpart of each instance is defined in an analogous way as in the previous setting.

To handle integrality, our cutting plane algorithm was embedded into the branch-and-cut framework provided by `IBM ILOG Cplex`, where, at each node of the branching tree, cuts are separated and added to the current formulation as global constraints. All integer solutions (including those provided by the solver's internal heuristics) are checked for robustness—through our separation procedure—before updating the incumbent, and are rejected in case they happen not to be robust.

Table 2 provides, for each problem, the same information as in Table 1. A time limit of 1,800 s was given for the solution of each instance. We removed 8 instances

Table 2 Results on robust MILP instances from MIPLIB 3.0 (`inf.` indicates that no feasible robust solution exists, `T.L.` indicates that the time limit has been reached)

	Instance								$\Gamma = 1$				$\Gamma = 10$				$\Gamma = 50$			
	n	m	z	T	m_u	p_u	n'	m'	$\%\Delta z$	t_{BS}	t_{CP}	#cuts	$\%\Delta z$	t_{BS}	t_{CP}	#cuts	$\%\Delta z$	t_{BS}	t_{CP}	#cuts
10teams	2,025	230	924.000	0.51	95	6,075	8,195	6,305	inf.	0.01	0.16	66	inf.	0.39	0.21	66	inf.	5.31	0.24	66
bell3a	133	123	878,430.316	1.78	123	347	603	470	4.774	1.10	0.43	100	8.446	2.05	0.73	61	8.446	2.05	0.74	61
bell5	104	91	8,966,406.492	0.11	91	266	461	357	inf.	0.01	0.01	60	inf.	0.01	0.00	49	inf.	0.01	0.00	49
blend2	353	274	7.599	1.03	185	1,065	1,603	1,339	inf.	0.02	0.01	18	inf.	0.04	0.02	19	inf.	0.06	0.02	19
cap6000	6,000	2,176	−2,451,294.000	0.33	2,053	47,947	56,000	50,123	92.972	0.84	954.54	5,676	92.976	80.99	46.14	1,998	92.992	109.65	59.06	2,102
dcmulti	548	290	188,186.500	0.31	212	660	1,420	950	inf.	0.01	0.03	195	inf.	0.03	0.05	182	inf.	0.04	0.05	182
egout	141	98	568.101	0.00	55	110	306	208	0.000	0.00	0.00	35	0.000	0.01	0.01	32	0.000	0.02	0.01	32
fixnet6	878	478	3,983.000	0.68	378	756	2,012	1,234	0.000	0.53	1.52	257	0.050	0.63	1.07	229	0.050	0.71	1.07	229
flugpl	18	18	1,201,500.000	0.00	12	30	60	48	0.993	0.01	0.02	12	1.130	0.01	0.01	9	1.130	0.02	0.01	9
gen	870	780	112,313.363	0.02	630	1,722	3,222	2,502	inf.	0.00	0.01	166	inf.	0.04	0.01	166	inf.	0.05	0.01	166
gesa2	1,224	1,392	25,779,856.372	0.29	1,344	4,920	7,488	6,312	20.539	5.39	610.41	1,225	26.158	420.97	1,060.95	1,070	26.158	303.65	`T.L.`	932
gesa3	1,152	1,368	27,991,042.648	0.58	1,320	4,800	7,272	6,168	22.703	3.53	`T.L.`	1,248	27.627	11.13	`T.L.`	903	27.627	1,457.26	`T.L.`	823
khb05250	1,350	101	106,940,226.000	0.06	24	48	1,422	149	0.472	0.10	0.11	44	0.637	0.19	0.11	24	0.637	0.19	0.12	24
l152lav	1,989	97	4,722.000	0.57	1	1,011	3,001	1,108	0.000	1.70	0.64	19	0.000	1.72	23.35	1,097	0.000	1.46	8.79	682
lseu	89	28	1,120.000	0.04	28	309	426	337	inf.	0.00	0.01	74	inf.	0.02	0.01	41	inf.	0.01	0.00	24
mod008	319	6	307.000	0.08	6	1,243	1,568	1,249	0.651	1.15	0.57	154	0.651	1.79	220.53	9,507	0.651	0.85	17.03	2,367
mod010	2,655	146	6,548.000	0.19	1	155	2,811	301	0.000	0.11	0.19	1	0.000	0.09	0.19	2	0.000	0.09	0.20	2
p0033	33	16	3,089.000	0.00	15	98	146	114	inf.	0.00	0.00	19	inf.	0.00	0.00	12	inf.	0.00	0.00	12
p0201	201	133	7,615.000	0.30	133	1,923	2,257	2,056	inf.	0.01	0.04	370	inf.	0.20	0.01	98	inf.	0.14	0.01	98
p0282	282	241	258,411.000	0.07	241	1,966	2,489	2,207	29.437	0.09	0.11	152	30.963	0.32	0.16	165	30.963	0.18	0.11	49
p0548	548	176	8,691.000	0.03	176	1,711	2,435	1,887	inf.	0.02	0.03	134	inf.	0.05	0.07	143	inf.	0.05	0.07	117

Table 2 continued

	Instance								Γ = 1				Γ = 10				Γ = 50			
	n	m	z	T	m_u	p_u	n'	m'	$\%\Delta z$	t_{BS}	t_{CP}	#cuts	$\%\Delta z$	t_{BS}	t_{CP}	#cuts	$\%\Delta z$	t_{BS}	t_{CP}	#cuts
p2756	2,756	755	3,124.000	0.27	755	8,937	12,448	9,692	230.314	3.24	T.L.	982	237.644	18.45	T.L.	1,086	inf.	0.34	2.38	426
pk1	86	45	11.000	102.31	30	60	176	105	1.010	135.34	39.64	54	2.020	41.15	31.39	30	2.020	43.22	31.36	30
pp08aCUTS	240	246	7,350.000	1.42	182	551	973	797	3.645	0.85	0.58	283	4.973	1.50	1.34	177	4.973	1.32	1.34	177
pp08a	240	136	7,350.000	0.55	72	192	504	328	2.482	0.34	0.29	103	2.656	1.89	0.42	72	2.656	1.11	0.42	72
qnet1	1,541	503	16,029.693	1.28	171	1,825	3,537	2,328	inf.	0.01	0.17	149	inf.	0.34	0.91	241	inf.	0.19	1.03	119
qnet1_o	1,541	456	16,029.693	1.01	124	1,417	3,082	1,873	inf.	0.01	0.13	95	inf.	0.17	1.19	270	inf.	0.11	0.25	77
rentacar	9,557	6,803	30,356,760.984	0.52	509	1,297	11,363	8,100	inf.	0.01	0.54	122	inf.	0.41	0.54	122	inf.	0.48	0.53	122
rgn	180	24	82.200	0.12	4	100	284	124	441.363	0.01	0.05	100	441.363	0.01	0.01	40	441.363	0.01	0.01	4
stein27	27	118	18.000	0.20	118	378	523	496	44.444	0.28	0.18	356	44.444	0.38	0.19	119	44.444	0.25	0.24	118
stein45	45	331	30.000	8.67	331	1,034	1,410	1,365	43.333	4.04	1.26	952	43.333	3.17	0.89	330	43.333	3.23	0.89	330
vpm1	378	234	20.000	0.00	192	504	1,074	738	15.000	0.03	0.03	57	15.000	0.05	0.08	54	15.000	0.05	0.07	54
vpm2	378	234	13.750	0.48	192	672	1,242	906	1.818	1.46	2.37	123	1.818	4.78	1.66	97	1.818	5.19	1.55	95
Arith. mean (all)				3.75						4.86	158.00	406.09		17.97	151.28	560.94		58.71	112.96	293.00

for which none of the two algorithms was able to compute, within the time limit, the optimal robust solution value for some value of Γ, thus obtaining a test set of 33 problems. The last line of the table provides the average computing time (arithmetic mean) of each algorithm, considering the time limit when an algorithm fails in solving an instance to optimality.

Our results show that, for the MIPLIB instances in our testbed, the cutting plane algorithm is outperformed by the compact formulation. Indeed, the cutting plane scheme is unable to solve 6 instances, and its typical computing time is significantly worse. As a matter of fact, there are several cases where the compact formulation requires shorter computing time even than the nominal problem—in which cases the cutting plane approach is clearly dominated by the compact formulation. In our view, this is due to two main reasons. First, a more powerful preprocessing and formulation tightening is achievable on the compact model that includes all robustness information; in addition, relevant information for branching, such as pseudocosts, are more reliable for the compact formulation. Second, the competitive advantage in solution speed of a cutting plane method is large at the root node, but is partly lost at the subsequent branching nodes where parametric reoptimization (dual simplex) is applied.

3 The uncertain set covering problem

We next move to a probabilistic setting, and introduce an uncertain version of the well-known Set Covering Problem (SCP). Given a 0–1 $m \times n$ matrix $A = (a_{ij})$ and an n-dimensional integer vector $c = (c_j)$ representing the cost of each column, the SCP requires to select a subset S of columns such that

- the sum of the costs of the selected columns is minimized;
- for each row i ($i = 1, \ldots, m$) there exists at least one column $j \in S$ such that $a_{ij} = 1$.

Set covering is a useful model for several important practical problems, and arises as a subproblem in many applications; see Caprara et al. [10] and Gamache et al. [17] for applications to railways and airline crew scheduling, or Balas [2] and Ceria et al. [11] for surveys on applications of SCP to location, routing and other problems. The huge amount of literature on SCP includes both exact and heuristic algorithms; a computational comparison of the main solution methods is reported in Caprara et al. [9].

Let $N = \{1, \ldots, n\}$ and $M = \{1, \ldots, m\}$ be the SCP column and row set, respectively. Moreover, for each $i \in M$ let $J_i = \{j \in N : a_{ij} = 1\}$ denote the set of columns covering row i. A straightforward ILP model for SCP is as follows:

$$\min \sum_{j \in N} c_j\, x_j \tag{13}$$

$$\text{s.t.} \quad \sum_{j \in J_i} x_j \geq 1, \quad i \in M, \tag{14}$$

$$x_j \in \{0, 1\}, \quad j \in N, \tag{15}$$

where each variable x_j takes value 1 if column j is selected, and 0 otherwise.

We next introduce a stochastic variant of SCP that we call the Uncertain Set Covering Problem (USCP), where coefficients a_{ij} in each column are random variables following a Bernoulli distribution. To be more specific, we assume each column $j \in N$ has an associated probability $p_j \in [0, 1[$ to disappear (i.e., that all coefficients in column j become zero), and each row $i \in M$ has an associated threshold $\overline{P_i} \in]0, 1]$ representing a minimum required probability for row i to be covered by at least one selected column. We assume that the random variables corresponding to column disappearance are independent. The problem requires to determine a set S of columns that minimizes objective function (13) and satisfies the ith constraint (14) with probability, at least, $\overline{P_i}$. USCP is strongly NP-hard, because it generalizes SCP. It arises in many practical problems, including crew scheduling applications where columns are associated with feasible pairings, and a column disappearing from the model corresponds to a nonshow of a crew.

A different SCP model that also deals with uncertainty is known in the literature as the Probabilistic Set Covering Problem (PSCP) [6], in which one wants to optimize over the set of binary vectors x such that $\mathbb{P}\{A\,x \geq \xi\} \geq p$, where ξ is a random binary right-hand side vector and p is a threshold given on input. PSCP was first studied by Beraldi and Ruszczyński [6], who proposed an exact algorithm based on the iterative solution of deterministic SCPs. Recently, Saxena et al. [35] reformulated PSCP as a mixed integer nonlinear program, linearized the corresponding model, and solved the resulting MIP with a general purpose commercial code.

Using again variables x_j $(j \in N)$, our USCP can be formulated as the following problem with chance constraints:

$$\min \sum_{j \in N} c_j\, x_j \tag{16}$$

$$\text{s.t.} \quad \mathbb{P}\{a_i^T\, x \geq 1\} \geq \overline{P_i}, \quad i \in M, \tag{17}$$

$$x_j \in \{0, 1\}, \quad j \in N. \tag{18}$$

Given a binary vector x, let us define $J_i(x) = \{j \in J_i : x_j = 1\}$ for $i \in M$. If x satisfies the ith nominal constraint (14), the probability that the associated uncertain constraint is violated is given by $\mathbb{P}\{a_i^T\, x < 1\} = \prod_{j \in J_i(x)} p_j$ (due to statistical independence). Indeed, since $x_j \in \{0, 1\}$ for all j and the random a_{ij}'s take values in $\{0, 1\}$ as well, the event $a_i^T x < 1$ is achieved when all columns corresponding to $x_j = 1$ disappear.

We then define as feasible only those solutions for which this probability is not larger than $1 - \overline{P_i}$, i.e., the solutions such that, for each row $i \in M$

$$\mathbb{P}\{a_i^T\, x < 1\} = \prod_{j \in J_i(x)} p_j \leq 1 - \overline{P_i} \tag{19}$$

Defining the nonnegative quantities $w_j = -\ln p_j$ $(j \in N)$ and $\overline{W_i} = -\ln(1 - \overline{P_i})$, the feasibility condition with respect to row i reads

$$\sum_{j \in J_i(x)} w_j \geq \overline{W_i}. \tag{20}$$

Without loss of generality we assume that, for each row i, the above condition is satisfied in case $J_i(x) = J_i$, since otherwise it would be impossible to cover row i with the required probability $\overline{P_i}$, hence USCP would be infeasible—despite the feasibility of the deterministic SCP (13)–(15).

3.1 Compact versus cutting plane USCP models

By using a modeling technique also exploited in Haight et al. [25], one easily obtains the following compact ILP model for USCP:

$$(\text{M1}) \quad \min \sum_{j \in N} c_j \, x_j \tag{21}$$

$$\text{s.t.} \quad \sum_{j \in J_i} w_j \, x_j \geq \overline{W_i}, \quad i \in M, \tag{22}$$

$$x_j \in \{0, 1\}, \quad j \in N. \tag{23}$$

The LP relaxation of the above model can be strengthened by exploiting the integrality of the x variables appearing in (22). As a matter of fact, due to the definition of the w_j's, these latter constraints often lead to knapsack conditions that are very challenging for MIP solvers. (Fortunately, as shown in the computational section, the rich arsenal of general-purpose preprocessing and cut generation procedures embedded in modern MIP solvers turns out to be quite effective for M1.) A first trivial strengthening is obtained by just replacing, for each row i, each coefficient w_j ($j \in J_i$) by $\min\{w_j, \overline{W_i}\}$. In addition, given a positive integer k, a simple rounding argument similar to that used to derive Gomory's fractional cuts allows one to derive the valid inequality

$$\sum_{j \in J_i} \left\lceil \frac{k-1}{\overline{W_i} - \epsilon} \, \overline{w}_j \right\rceil x_j \geq \left\lceil \frac{k-1}{\overline{W_i} - \epsilon} \, \overline{W_i} \right\rceil = k \tag{24}$$

where $k \geq 2$ is an integer parameter giving the desired right-hand side value, and ϵ is a small positive value. Because of their combinatorial nature, constraints (24) are numerically more stable than (22). However, it is easy to see that no LP dominance between (22) and (24) exists, hence we keep constraints (22) in our compact model, while constraints (24) are treated as additional cuts possibly used to strengthen the formulation.

A cutting plane model for USCP can instead be derived from (16)–(18) by replacing the ith covering constraint with its "uncertain" counterpart

$$\sum_{j \in J_i} x_j - \beta(x, \overline{W_i}) \geq 1, \tag{25}$$

where term $\beta(x, \overline{W_i})$ represents the maximum decrease of the left-hand side associated with uncertain situations we want to take care of. In our problem, these situations are those that can arise with a probability larger than $1 - \overline{P_i}$.

Constraints (25) can now be modeled in a linear way, so to obtain the following (noncompact) ILP:

$$\text{(M2)} \quad \min \sum_{j \in N} c_j \, x_j \tag{26}$$

$$\text{s.t.} \quad \sum_{j \in J_i} x_j - \sum_{j \in S} x_j \geq 1, \quad S \subseteq J_i : \sum_{j \in S} w_j < \overline{W_i}, \; i \in M, \tag{27}$$

$$x_j \in \{0, 1\}, \quad j \in N. \tag{28}$$

Constraints (27) impose that each row $i \in M$ must be covered by a subset of columns having a small probability to disappear all together.

The exponential number of constraints in model M2 immediately suggests to adopt a cutting plane algorithm where constraints (27) are added to the formulation on the fly, when they are needed. Given the current (possibly fractional) solution x^*, the separation problem associated with a given row i requires to find (if any) a set $S \subseteq J_i$ such that $\sum_{j \in S} w_j < \overline{W_i}$ and $\sum_{j \in S} x^*_j > \sum_{j \in J_i} x^*_j - 1$, and can be solved through the following ILP:

$$\beta(x^*, \overline{W_i}) = \max \sum_{j \in J_i} x^*_j \, d_{ij} \tag{29}$$

$$\text{s.t.} \quad \sum_{j \in J_i} w_j \, d_{ij} \leq \overline{W_i} - \epsilon, \tag{30}$$

$$d_{ij} \in \{0, 1\}, \; j \in J_i, \tag{31}$$

where ϵ is a sufficiently small positive value. Then, let d^* be an optimal solution to the above separation ILP, and define $S^* = \{j \in J_i : d^*_{ij} = 1\}$. If $\sum_{j \in S^*} x^*_j$ $(= \beta(x^*, \overline{W_i})) > \sum_{j \in J_i} x^*_j - 1$, then constraint (27) associated with row i and set S^* is violated by x^* and can be added to the current formulation.

Validity of formulation M2 is stated in the following theorem.

Theorem 1 *Let $F(M1) := \{x \in \{0, 1\}^n : (22)$ hold$\}$ and $F(M2) = \{x \in \{0, 1\}^n : (27)$ hold$\}$ denote the set of feasible solutions to M1 and to M2, respectively. Then $F(M1) = F(M2)$.*

Proof We first prove that any feasible solution x^* to M1 satisfies constraint (25) for any $i \in M$. As x^* is feasible for M1, we have $\sum_{j \in J_i} w_j x^*_j \geq \overline{W_i}$, hence constraint (30) imposes that not all d_{ij} variables with $x^*_{ij} = 1$ can be set to 1. This implies $\beta(x^*, \overline{W_i}) \leq \sum_{j \in J_i} x^*_j - 1$, i.e., constraint (25) is satisfied by x^*.

Conversely we prove that, for any feasible solution x^* to M2, all constraints (22) are satisfied. Assume, by contradiction, that x^* violates constraint (22) for a certain $i \in M$, i.e., $\sum_{j \in J_i} w_j \, x^*_j < \overline{W_i}$. An optimal solution to the separation problem

(29)–(31) is therefore $d^* = x^*$, thus $\beta(x^*, \overline{W_i}) = \sum_{j \in J_i} x_j^*$, which makes constraint (25) violated, a contradiction. □

The link between formulations M1 and M2 is made more evident by observing that the knapsack inequalities (22) actually model conditions (20), and imply the knapsack cover inequalities (27); see, e.g., [30]. In this view, the equivalence between M1 and M2 can also be derived from a well-known property of knapsack problems, namely, that the family of cover inequalities (together with the binary restriction on the variables) suffices to provide a valid knapsack formulation.

Theorem 1 states that the sets of feasible solutions to M1 and M2 coincide. However, as already mentioned, this equivalence does not necessarily hold when solutions to the LP relaxations of the two models are considered. The following example shows that no dominance exists among the optimal solution value of the LP relaxation of M1 and the optimal solution value of the LP relaxation of model M2. Consider the SCP instance with $m = 1$, $n = 3$, $c_j = 1$, $w_j = 3$ $(j = 1, \dots, 3)$ and $\overline{W_1} = 5$. The LP relaxation of M1 yields solution $x_1 = x_2 = x_3 = 5/9$, with a lower bound equal to $5/3$, while an optimal solution for the LP relaxation of M2 is $x_1 = x_2 = x_3 = 1/2$, having value $3/2$. When considering the same instance with $\overline{W_1} = 4$, the optimal solution of the LP relaxation of M2 is the same, while the optimal solution of the LP relaxation of M1 is $x_1 = x_2 = x_3 = 4/9$, having value $4/3$ (i.e., worse than the value of the LP relaxation of M2).

3.2 Computational experiments

In this section we report our experiments to determine the best practical method for the exact solution of USCP instances. We first describe a heuristic algorithm that is used to provide an initial feasible solution, and then we give computational experiments on a large set of instances from the literature.

3.2.1 A heuristic algorithm

We implemented an ad-hoc heuristic procedure aimed at producing an initial upper bound value. A classical heuristic algorithm for SCP is the greedy algorithm: start from a partial (possibly empty) solution and add the "best" column (according to certain "scores") to the current solution, halting the execution when the current solution is feasible. Different methods have been proposed in the literature to compute the score of each column. From a computational viewpoint, one of the most effective criteria is to define the score of each column j with respect to a partial solution as the ratio between the cost of column j and the number of currently uncovered rows that can be covered by column j. At the end of the algorithm, a post optimization procedure that removes unnecessary columns can be applied in case some rows are overcovered. Although the solutions produced by the greedy algorithm can be far from optimality, the required computational effort is generally negligible.

Our heuristic for USCP has a similar philosophy, in that it is a greedy approach that starts from a partial solution and iteratively adds the "best" column to the current solution, until the solution becomes feasible. Given a current solution, we check feasibility by scanning constraints (22). If the current solution is feasible, the algorithm stops. Otherwise, a minimum-cost unselected column appearing in the first violated constraint is added to the solution. The algorithm is executed several times, starting from different partial solutions, so as to produce different outcomes, and returns the best solution found to be used as initial upper bound. In our implementations, we start with

- an empty solution, or
- the heuristic solution found by the greedy algorithm for the nominal problem, or
- the heuristic solution found by the greedy algorithm for the nominal problem restricted to the columns with $w_j \geq \max_i\{\overline{W}_i : a_{ij} = 1\}$, with high-cost slack variables added for each constraint so as to ensure feasibility. Note that if no slack variables are used by the greedy, the associated solution turns out to be feasible for USCP as well; for this reason, this method is likely to be quite effective for problems with small $\overline{P}_i$'s.

3.2.2 Computational results

In our experiments, we considered the set covering instances publicly available at the ORLIB library [33], and tested our ILP models on the 60 easiest ones, namely those that could be solved to proven optimality within 600 CPU seconds using `IBM ILOG Cplex` 11.0 on our computer (PC Intel Q6600 CPU@2.40 GHz, 4GB RAM).

Starting from these instances, we set the threshold probability associated with each row $i \in M$ as $\overline{P_i} = P^{\mathtt{min}}$, where $P^{\mathtt{min}}$ is a parameter. Probabilities p_j associated with columns $j \in N$ were randomly generated according to a uniform distribution in [0, 0.2]. As to the definition of coefficients w_j (resp., $\overline{W_i}$), we computed the logarithm of each p_j (resp., $\overline{P_i}$), multiplied it by 1,000 and rounded the result to the nearest integer.

Tables 3 and 4 report the outcome of our experiments for different values of $P^{\mathtt{min}} \in \{0.85, 0.90, 0.95, 0.99\}$. In particular, for each instance we report the following information:

- optimal value (z^*) and solution time (T) of the nominal problem;
- best solution found (z_h) by our initial heuristic, along with the associated computing time (T_h),
- optimal value (z_u) of the uncertain version of the problem (an asterisk indicates that the optimal value is not known),
- computing time (T_{M1}) for model M1,
- computing time ($T_{\mathrm{M1}'}$) for model M1′, i.e., model M1 when constraints (24) for $k = 2$ and $k = 3$ are added to the initial formulation as "lazy constraints",
- overall computing time (T_{M2}) for model M2 and the associated separation time (T_{sep}). In our implementation, the separation problem (29)–(31) is solved for each row i at each node of the branch-and-bound tree. In case the current solution is noninteger, separation is carried out by solving a 0–1 knapsack problem by means of algorithm *combo* by Martello et al. [29], which is considered one of the most effective codes for solving knapsack problems. On the contrary, if the current

Table 3 Results on uncertain set covering instances for $P^{\mathtt{min}} \in \{0.85, 0.90\}$

Nominal problem			$P^{\mathtt{min}} = 0.85$							$P^{\mathtt{min}} = 0.90$						
Name	z^*	T	z_h	T_h	z_u	T_{M1}	$T_{M1'}$	T_{M2}	T_{sep}	z_h	T_h	z_u	T_{M1}	$T_{M1'}$	T_{M2}	T_{sep}
scp41	429	0.01	644	0.01	592	0.02	0.03	0.03	0.01	814	0.01	701	0.04	0.05	0.04	0.00
scp42	512	0.01	663	0.01	612	0.03	0.04	0.03	0.00	918	0.01	782	0.04	0.06	0.10	0.00
scp43	516	0.00	758	0.01	662	0.05	0.06	0.12	0.00	930	0.01	830	0.19	0.24	0.46	0.01
scp44	494	0.02	678	0.01	602	0.03	0.05	0.03	0.00	822	0.01	740	0.12	0.15	0.26	0.01
scp45	512	0.00	702	0.01	656	0.09	0.12	0.13	0.00	912	0.01	797	0.15	0.22	0.39	0.00
scp46	560	0.01	710	0.01	655	0.02	0.04	0.02	0.00	903	0.01	796	0.07	0.09	0.29	0.02
scp47	430	0.01	595	0.00	579	0.03	0.05	0.03	0.00	778	0.01	695	0.03	0.07	0.05	0.00
scp48	492	0.03	779	0.01	647	0.06	0.08	0.17	0.00	1,020	0.00	841	0.29	0.15	0.52	0.02
scp49	641	0.02	911	0.01	809	0.06	0.08	0.13	0.00	1,127	0.01	984	0.11	0.40	0.62	0.00
scp410	514	0.01	791	0.01	708	0.05	0.08	0.09	0.00	911	0.01	811	0.03	0.06	0.11	0.01
scp51	253	0.05	339	0.01	302	0.09	0.10	0.10	0.00	461	0.01	391	0.16	0.17	0.56	0.03
scp52	302	0.10	423	0.02	371	0.17	0.15	0.20	0.00	541	0.02	459	0.93	0.98	1.19	0.06
scp53	226	0.01	331	0.02	300	0.03	0.04	0.03	0.00	404	0.02	344	0.05	0.06	0.10	0.00
scp54	242	0.03	365	0.02	321	0.07	0.08	0.11	0.00	442	0.01	370	0.08	0.11	0.24	0.01
scp55	211	0.02	321	0.02	300	0.07	0.05	0.04	0.00	378	0.01	350	0.11	0.07	0.15	0.00
scp56	213	0.01	301	0.02	291	0.05	0.07	0.10	0.00	406	0.01	369	0.08	0.09	0.24	0.01
scp57	293	0.05	413	0.02	386	0.10	0.08	0.08	0.00	562	0.02	480	0.23	0.23	0.40	0.03
scp58	288	0.04	407	0.02	351	0.07	0.10	0.12	0.00	494	0.02	415	0.08	0.10	0.35	0.03
scp59	279	0.03	419	0.02	370	0.05	0.06	0.04	0.00	544	0.01	477	0.49	0.44	0.86	0.03
scp510	265	0.01	404	0.01	360	0.16	0.11	0.14	0.01	455	0.01	429	0.11	0.16	0.23	0.00
scp61	138	0.32	162	0.01	153	0.18	0.19	0.16	0.00	218	0.01	199	0.52	0.50	0.59	0.01
scp62	146	0.35	199	0.01	174	0.47	0.44	0.33	0.00	235	0.01	199	0.48	0.61	0.48	0.03
scp63	145	0.25	212	0.01	169	0.21	0.18	0.18	0.00	279	0.01	218	0.65	0.74	1.11	0.08

Table 3 continued

Nominal problem			$P^{\min} = 0.85$							$P^{\min} = 0.90$						
Name	z^*	T	z_h	T_h	z_u	T_{M1}	$T_{M1'}$	T_{M2}	T_{sep}	z_h	T_h	z_u	T_{M1}	$T_{M1'}$	T_{M2}	T_{sep}
scp64	131	0.03	190	0.01	157	0.14	0.16	0.19	0.00	249	0.01	196	0.63	0.77	0.61	0.05
scp65	161	0.44	252	0.01	219	0.80	0.78	0.72	0.05	301	0.01	251	1.22	0.92	1.81	0.20
scpa1	253	0.98	308	0.03	291	0.27	0.27	0.32	0.00	412	0.04	385	1.47	1.96	5.26	0.34
scpa2	252	0.71	349	0.03	300	0.15	0.20	0.22	0.00	411	0.04	368	0.77	0.65	1.21	0.07
scpa3	232	0.48	306	0.03	278	0.51	0.37	0.42	0.00	391	0.04	343	0.31	0.44	1.12	0.03
scpa4	234	0.29	306	0.02	284	0.62	0.40	0.41	0.01	420	0.04	359	1.69	1.90	4.18	0.26
scpa5	236	0.10	340	0.03	312	0.58	0.51	0.50	0.00	419	0.04	359	0.68	0.87	1.84	0.08
scpb1	69	0.96	111	0.03	97	2.86	5.01	5.67	0.64	123	0.03	109	11.70	16.23	65.22	5.60
scpb2	76	1.58	101	0.03	89	3.01	8.63	2.98	0.25	123	0.03	101	4.19	11.94	24.22	2.61
scpb3	80	0.89	113	0.03	100	2.50	6.06	2.96	0.25	141	0.03	125	20.59	103.51	187.28	11.64
scpb4	79	1.95	108	0.03	91	1.50	1.28	1.03	0.10	123	0.04	109	10.32	12.41	72.99	8.10
scpb5	72	0.80	99	0.02	88	1.89	2.22	3.24	0.18	118	0.03	99	1.46	1.45	1.74	0.23
scpc1	227	0.81	323	0.06	287	1.65	1.72	1.26	0.01	427	0.06	360	5.23	9.69	32.11	1.89
scpc2	219	1.58	329	0.05	275	2.71	3.13	7.29	0.40	396	0.06	340	6.88	17.95	343.21	18.98
scpc3	243	1.71	356	0.05	310	2.52	3.59	2.64	0.15	434	0.05	373	17.21	12.39	87.62	4.09
scpc4	219	0.93	307	0.05	275	1.97	3.26	1.80	0.07	358	0.05	328	3.25	4.52	11.40	0.67
scpc5	215	0.86	312	0.05	278	1.30	2.34	1.60	0.06	375	0.05	333	4.56	8.39	68.63	4.09
scpd1	60	1.50	85	0.05	76	10.89	12.63	10.80	1.43	98	0.06	89	6.46	19.05	43.57	4.57
scpd2	66	6.96	96	0.05	82	15.49	37.83	28.27	3.06	111	0.06	96	18.35	32.21	331.71	25.36
scpd3	72	6.68	106	0.05	90	58.20	91.42	204.43	20.45	116	0.06	100	26.06	82.10	346.95	26.06
scpd4	62	3.18	85	0.06	78	10.66	10.87	10.30	1.08	105	0.06	91	44.68	48.75	700.73	61.09

Table 3 continued

Nominal problem			$P^{\mathtt{min}} = 0.85$							$P^{\mathtt{min}} = 0.90$						
Name	z^*	T	z_h	T_h	z_u	T_{M1}	$T_{M1'}$	T_{M2}	T_{sep}	z_h	T_h	z_u	T_{M1}	$T_{M1'}$	T_{M2}	T_{sep}
scpd5	61	0.98	86	0.05	81	10.29	12.64	21.55	2.73	99	0.05	91	4.62	7.81	22.73	2.94
scpe1	5	0.26	5	0.00	5	3.18	0.19	23.70	0.36	5	0.00	5	1.43	2.82	42.76	0.55
scpe2	5	0.27	6	0.00	5	0.25	0.22	0.69	0.00	7	0.00	5	0.15	0.17	1.36	0.03
scpe3	5	0.29	8	0.00	5	0.27	0.33	0.76	0.00	8	0.00	5	0.15	0.12	1.08	0.00
scpe4	5	0.26	7	0.00	5	0.27	0.25	0.64	0.01	8	0.00	5	0.17	0.18	1.18	0.02
scpe5	5	0.26	5	0.00	5	0.28	4.32	0.55	0.00	5	0.00	5	2.08	0.16	1.58	0.01
scpnre1	29	64.69	42	0.08	33	121.12	541.78	474.74	46.44	46	0.09	40	1,743.26	T.L.	T.L.	116.76
scpnre2	30	232.01	40	0.08	32	71.55	102.52	214.09	21.53	45	0.09	38	658.24	T.L.	T.L.	120.76
scpnre3	27	73.86	41	0.08	31	71.77	361.34	368.96	39.11	44	0.09	35	146.83	370.09	T.L.	130.65
scpnre4	28	77.92	42	0.08	33	155.02	298.65	T.L.	193.47	48	0.10	38	1,587.92	T.L.	T.L.	122.70
scpnre5	28	28.71	42	0.08	32	28.01	51.52	104.55	11.51	44	0.09	40[a]	T.L.	T.L.	T.L.	154.80
scpnrf1	14	49.95	20	0.10	14	15.33	29.95	40.73	9.24	21	0.09	16	40.71	117.60	1,546.97	147.69
scpnrf2	15	39.89	20	0.08	16	23.14	32.38	94.50	20.30	22	0.09	19	172.69	669.52	T.L.	154.84
scpnrf3	14	25.92	19	0.09	15	14.77	20.71	59.78	12.86	23	0.09	19	201.25	1,008.18	T.L.	139.66
scpnrf4	14	93.99	18	0.10	15	78.35	226.32	804.98	109.03	22	0.09	17	185.67	676.83	T.L.	123.91
scpnrf5	13	372.68	20	0.08	14	284.48	987.07	T.L.	252.37	23	0.09	15	63.41	144.03	T.L.	144.92
Arith. mean		18.28		0.03		16.67	47.75	101.56	12.45		0.04		113.36	176.52	336.01	25.61
# opt.		60				60	60	58					59	56	51	

[a] Optimal value is not known

Table 4 Results on uncertain set covering instances for $P^{\mathtt{min}} \in \{0.95, 0.99\}$

Nominal problem			$P^{\mathtt{min}} = 0.95$							$P^{\mathtt{min}} = 0.99$						
Name	z^*	T	z_h	T_h	z_u	T_{M1}	$T_{M1'}$	T_{M2}	T_{sep}	z_h	T_h	z_u	T_{M1}	$T_{M1'}$	T_{M2}	T_{sep}
scp41	429	0.01	1,026	0.01	921	0.06	0.12	0.59	0.03	1,578	0.01	1,421	0.67	0.50	T.L.	78.94
scp42	512	0.01	1,156	0.01	983	0.11	0.15	1.94	0.08	1,545	0.01	1,379	0.77	0.60	1,360.97	48.54
scp43	516	0.00	1,112	0.01	1,048	1.30	1.39	29.01	2.65	1,838	0.01	1,526	1.64	1.43	T.L.	72.49
scp44	494	0.02	1,106	0.01	977	0.36	0.30	3.59	0.40	1,588	0.01	1,388	0.60	0.37	124.32	4.93
scp45	512	0.00	1,154	0.01	1,065	0.81	0.67	5.45	0.37	1,616	0.01	1,429	0.93	0.58	389.81	13.55
scp46	560	0.01	1,235	0.01	1,113	0.50	0.41	25.34	1.62	1,700	0.01	1,522	2.47	2.17	T.L.	41.88
scp47	430	0.01	1,046	0.01	975	0.07	0.21	1.41	0.11	1,450	0.01	1,337	0.69	0.77	649.03	31.85
scp48	492	0.03	1,181	0.01	1,054	0.54	0.51	4.19	0.30	1,758	0.01	1,462	1.38	1.57	T.L.	69.84
scp49	641	0.02	1,323	0.01	1,222	1.11	1.49	238.35	11.71	2,027	0.01	1,647	1.46	0.95	T.L.	55.14
scp410	514	0.01	1,233	0.01	1,117	0.28	0.25	1.51	0.10	1,813	0.01	1,638	2.47	2.86	T.L.	61.06
scp51	253	0.05	520	0.02	467	0.58	0.39	5.69	0.44	807	0.02	682	0.84	0.54	T.L.	64.83
scp52	302	0.10	656	0.02	567	0.66	0.73	252.73	16.29	921	0.02	786	2.23	1.37	T.L.	49.84
scp53	226	0.01	537	0.01	446	0.04	0.06	0.70	0.03	818	0.02	696	1.31	1.29	T.L.	92.09
scp54	242	0.03	550	0.02	466	0.33	0.33	11.08	1.06	735	0.02	672	1.81	1.11	T.L.	66.43
scp55	211	0.02	466	0.01	425	0.15	0.15	0.59	0.03	720	0.02	640	0.65	0.50	246.64	16.78
scp56	213	0.01	517	0.02	465	0.44	0.32	3.96	0.32	829	0.02	723	3.27	4.39	T.L.	52.86
scp57	293	0.05	691	0.02	597	0.13	0.21	7.57	0.72	1,006	0.02	873	1.81	1.66	T.L.	58.70
scp58	288	0.04	584	0.02	527	0.40	0.36	27.43	2.92	815	0.02	753	1.83	1.40	T.L.	62.40
scp59	279	0.03	674	0.02	578	1.40	1.11	28.48	1.33	979	0.02	823	3.41	2.52	T.L.	51.61
scp510	265	0.01	586	0.02	537	0.88	0.56	5.63	0.44	881	0.02	762	1.41	0.62	T.L.	56.60
scp61	138	0.32	255	0.01	236	0.74	0.75	7.88	0.78	395	0.01	340	2.29	2.79	T.L.	56.51
scp62	146	0.35	283	0.01	253	1.12	1.06	25.19	2.57	371	0.01	341	2.34	1.65	1,061.53	33.96
scp63	145	0.25	326	0.01	275	1.56	1.63	69.56	3.90	438	0.01	370	9.06	8.01	T.L.	40.02

Table 4 continued

Nominal problem			$P^{\min} = 0.95$							$P^{\min} = 0.99$						
Name	z^*	T	z_h	T_h	z_u	T_{M1}	$T_{M1'}$	T_{M2}	T_{sep}	z_h	T_h	z_u	T_{M1}	$T_{M1'}$	T_{M2}	T_{sep}
scp64	131	0.03	286	0.01	235	1.11	1.27	4.59	0.31	385	0.01	338	6.07	3.92	T.L.	56.66
scp65	161	0.44	321	0.01	311	1.31	1.35	144.22	9.17	479	0.01	441	28.45	29.13	T.L.	35.86
scpa1	253	0.98	524	0.04	472	3.71	6.60	398.47	20.43	781	0.05	676	67.10	102.76	T.L.	29.04
scpa2	252	0.71	504	0.04	452	2.20	3.10	446.15	34.23	769	0.04	661	7.48	4.01	T.L.	43.97
scpa3	232	0.48	490	0.04	442	2.86	3.03	T.L.	105.27	727	0.04	630	21.94	20.80	T.L.	36.00
scpa4	234	0.29	500	0.03	448	4.89	5.68	T.L.	94.87	723	0.04	617	26.17	19.28	T.L.	40.15
scpa5	236	0.10	519	0.04	447	1.92	2.33	T.L.	120.09	750	0.05	658	16.23	16.19	T.L.	34.95
scpb1	69	0.96	133	0.03	125	8.45	4.24	T.L.	106.66	200	0.04	176	423.98	474.54	T.L.	37.01
scpb2	76	1.58	155	0.03	118	4.31	7.54	300.83	20.50	196	0.04	167	202.08	263.30	T.L.	48.42
scpb3	80	0.89	156	0.03	141	14.10	29.34	T.L.	106.79	209	0.03	191	55.02	64.39	T.L.	36.28
scpb4	79	1.95	154	0.03	135	27.77	75.84	T.L.	106.88	208	0.04	185	67.66	77.62	T.L.	32.33
scpb5	72	0.80	140	0.03	122	2.46	4.42	207.13	17.22	197	0.04	170	41.27	47.42	T.L.	43.43
scpc1	227	0.81	507	0.06	442	39.27	35.76	T.L.	72.53	695	0.07	611	T.L.	1,296.43	T.L.	29.38
scpc2	219	1.58	466	0.06	404	40.08	47.82	T.L.	71.43	663	0.07	583[a]	T.L.	T.L.	T.L.	28.68
scpc3	243	1.71	540	0.06	455	107.92	45.11	T.L.	68.39	710	0.06	631[a]	T.L.	T.L.	T.L.	30.38
scpc4	219	0.93	461	0.06	417	103.58	148.79	T.L.	74.58	631	0.06	561	188.19	172.16	T.L.	24.43
scpc5	215	0.86	482	0.06	422	44.53	53.19	T.L.	71.30	659	0.07	565	40.66	27.47	T.L.	27.76
scpd1	60	1.50	116	0.06	106	26.98	34.04	T.L.	120.02	164	0.06	145	380.53	992.55	T.L.	40.12
scpd2	66	6.96	127	0.06	116	105.56	182.26	T.L.	109.09	176	0.07	157[a]	T.L.	T.L.	T.L.	45.82
scpd3	72	6.68	137	0.06	124	175.48	503.07	T.L.	90.20	191	0.07	164[a]	T.L.	T.L.	T.L.	39.68
scpd4	62	3.18	125	0.06	108	47.54	62.05	T.L.	111.91	164	0.06	148	T.L.	1,598.90	T.L.	41.18
scpd5	61	0.98	131	0.07	118	40.18	68.26	T.L.	122.76	179	0.06	156	172.89	483.81	T.L.	45.59

Table 4 continued

Nominal problem			$P^{\mathtt{min}} = 0.95$							$P^{\mathtt{min}} = 0.99$						
Name	z^*	T	z_h	T_h	z_u	T_{M1}	$T_{M1'}$	T_{M2}	T_{sep}	z_h	T_h	z_u	T_{M1}	$T_{M1'}$	T_{M2}	T_{sep}
scpe1	5	0.26	10	0.00	6	0.19	0.29	27.30	0.17	13	0.00	8	0.60	0.90	T.L.	3.33
scpe2	5	0.27	7	0.00	6	0.25	0.62	46.63	0.31	9	0.00	7	0.72	0.20	T.L.	3.69
scpe3	5	0.29	9	0.00	5	0.04	0.03	0.96	0.03	12	0.00	7	0.21	0.25	T.L.	3.67
scpe4	5	0.26	10	0.00	6	0.25	0.62	39.50	0.49	13	0.00	7	0.65	0.85	T.L.	3.15
scpe5	5	0.26	9	0.00	6	0.28	1.15	55.31	0.52	12	0.00	7	0.20	0.23	T.L.	3.33
scpnre1	29	64.69	54	0.09	45	485.43	T.L.	T.L.	96.62	65	0.11	58[a]	T.L.	T.L.	T.L.	50.83
scpnre2	30	232.01	49	0.09	44	524.58	T.L.	T.L.	110.34	64	0.10	56[a]	T.L.	T.L.	T.L.	55.53
scpnre3	27	73.86	50	0.09	40	83.72	391.18	T.L.	104.90	61	0.10	52[a]	T.L.	T.L.	T.L.	50.80
scpnre4	28	77.92	54	0.10	44	736.23	1,709.17	T.L.	103.18	64	0.10	56[a]	T.L.	T.L.	T.L.	54.38
scpnre5	28	28.71	52	0.09	46	1,415.73	T.L.	T.L.	113.75	72	0.10	58[a]	T.L.	T.L.	T.L.	58.71
scpnrf1	14	49.95	21	0.09	19	40.97	118.53	T.L.	192.37	26	0.10	23	245.05	451.10	T.L.	80.97
scpnrf2	15	39.89	24	0.10	21	46.89	125.33	T.L.	145.36	31	0.10	26	492.66	558.23	T.L.	61.93
scpnrf3	14	25.92	24	0.08	21	73.47	231.84	T.L.	133.67	30	0.11	26	627.35	1,245.31	T.L.	77.08
scpnrf4	14	93.99	24	0.09	20	304.69	1,037.93	T.L.	139.41	30	0.12	24	1,659.12	T.L.	T.L.	72.94
scpnrf5	13	372.68	22	0.10	18	453.80	586.95	T.L.	155.60	29	0.10	23[a]	T.L.	T.L.	T.L.	73.98
Arith. mean		18.28		0.04		83.11	182.36	820.48	49.99		0.04		440.29	463.19	1,683.87	44.37
# opt.		60				60	57	34					48	49	6	

[a] Optimal value is not known

solution is integer, separation can be carried out in linear time, by selecting d_{ij} variables according to increasing weights, until the maximum threshold $\overline{W_i}$ is reached.

To avoid an incorrect update of the incumbent by solver's internal heuristics, all integer solutions that do not pass our robustness check are rejected. A time limit of 1,800 CPU seconds was given to each model. The last rows of each table report, for each solution method, the average computing time, in seconds (arithmetic mean; for the unsolved instances, the time limit is considered) along with the number of instances solved to proven optimality within the given time limit.

Computational results show that model M1 qualifies as the best method to solve our USCP instances. This compact formulation is able to solve all instances but 13, with a reasonable average computing time. Note that increasing the value of $P^{\mathtt{min}}$ from 0.85 to 0.99 leads to increasingly difficult problems and, as expected, to worse solution values—making the associated problem more constrained.

As to the cutting plane model M2, it is not able to solve 91 instances out of 240, although the separation procedure itself is quite efficient, as it requires less than 10 % of the overall computing time, on average. However, as already observed in the previous section, for integer programs the cutting plane approach has the main disadvantage of hiding some relevant information on the solution structure, thus weakening powerful ingredients of modern MIP solvers such as preprocessing, branching on pseudocosts, cut generation, symmetry detection, etc. As a result, the compact formulation is typically preferable for USCP.

Quite surprisingly, the addition of constraints (24) in M1′ did not improve the overall performance—actually, it produced a certain slow-down. This behavior is due not only to the extra time required to handle the additional cuts, but also to the effectiveness of the general-purpose preprocessing and cut generation procedures embedded in `IBM ILOG Cplex`. Indeed, these procedures were able to squeeze automatically most of the information conveyed by the additional cuts (24).

To quantify the impact of preprocessing and cut generation on the performance of models M1, M1′ and M2, we deactivated these features from the `IBM ILOG Cplex` solver and reran our tests. Table 5 gives a summary of the outcome of this experiment and reports, for each value of $P^{\mathtt{min}}$, the average computing time (arithmetic mean) and the number of instances solved to proven optimality when using models M1, M1′ and M2 with and without preprocessing and cut generation, respectively. According to the table, the ranking among the three models radically changes when preprocessing and cuts are deactivated. As expected, the performance of M2 is almost unchanged, whereas for both M1 and M1′ it deteriorates significantly. In the new setting, model M2 has a better performance than model M1′, the latter clearly outperforming model M1.

4 The uncertain graph connectivity problem

In this section we consider an application of USCP to the uncertain counterpart of a graph connectivity problem.

Table 5 Average results on uncertain set covering instances for models M1, M1′ and M2 with and without preprocessing and cut generation, respectively

$p^{\mathtt{min}}$	With preprocessing and cuts						Without preprocessing and cuts					
	M1		M1′		M2		M1		M1′		M2	
	Time	#opt	Time	#opt	Time	#opt	Time	#opt	Time	#opt	Time	#opt
0.85	16.67	60	47.75	60	101.56	58	1,361.82	17	391.50	55	101.55	58
0.90	113.36	59	176.52	56	336.01	51	1,634.25	6	849.60	38	335.86	51
0.95	83.11	60	182.36	57	820.48	34	1,650.09	5	814.44	38	820.48	34
0.99	440.29	48	463.19	49	1,683.87	6	1,650.51	5	1,643.97	6	1,683.87	6

Given an undirected graph $G = (V, E)$ with nonnegative edge costs c_e, the classical Minimum Spanning Tree Problem (MSTP) requires to find a minimum-cost set of edges so that the associated subgraph is connected. By associating each edge $e \in E$ with a binary variable x_e taking value 1 iff edge e is selected, a set covering model for MSTP is as follows:

$$\min \sum_{e \in E} c_e \, x_e \tag{32}$$

$$\text{s.t.} \quad \sum_{e \in \delta(S)} x_e \geq 1, \quad S \subset V, \; S \neq \emptyset, \tag{33}$$

$$x_e \in \{0, 1\}, \quad e \in E. \tag{34}$$

In many practical applications arising in communications networks, survivability of the network is a major issue. Hence, one has to take care of possible link failures, which is often modeled by solving a variant of MSTP in which the set of selected edges must provide (at least) k edge-disjoint paths between each pair of nodes, k being an input parameter; see, e.g., Monma et al. [31], Grötschel and Monma [23], and Grötschel et al. [24].

Alternatively, and perhaps more realistically, possible failures in the connections can be modeled by associating each edge $e \in E$ with a failure probability $p_e \in [0, 1[$ and requiring a minimum probability $\overline{P} \in]0, 1]$ for connectivity. A same reasoning as for USCP shows that, assuming statistical independence of the random variables corresponding to edge failures, an uncertain version of the problem can be stated as:

$$(\text{M1}_\text{G}) \quad \min \sum_{e \in E} c_e \, x_e \tag{35}$$

$$\text{s.t.} \quad \sum_{e \in \delta(S)} w_e x_e \geq \overline{W}, \; S \subset V, \; S \neq \emptyset, \tag{36}$$

$$x_e \in \{0, 1\}, \quad e \in E, \tag{37}$$

where $w_e = -\ln(p_e)$ $(e \in E)$ and $\overline{W} = -\ln(1 - \overline{P})$.

Despite the deterministic version of the problem is polynomially solvable, our uncertain graph connectivity problem turns out to be strongly NP-hard, as stated by the following theorem.

Theorem 2 *The uncertain graph connectivity problem is strongly NP-hard.*

Proof We prove NP-hardness of our problem by reduction from the min-cost k-edge connected subgraph problem (kECSP), which requires to find a k-edge connected subgraph for a given undirected and weighted graph $G = (V, E)$. kECSP is known to be NP-hard also in case $k = 2$ [18], and remains so even when edge costs satisfy the triangle inequality. Indeed, finding a Hamiltonian cycle on an undirected graph $G = (V, E)$ corresponds to finding a 2ECSP solution having cost equal to n on a complete graph with $n = |V|$ nodes and costs equal to 1 for edges in E and 2 otherwise.

Given a 2ECSP instance, we define an instance of uncertain graph connectivity as follows: each edge has probability 1/2 of disappearing and the required probability for connection is equal to 3/4. This imposes that at least two edges are selected for each cut, and makes the set of feasible solutions coincide with the set of 2ECSP feasible solutions. □

The exponential number of constraints (36) makes it natural to use a branch-and-cut algorithm based on cutting planes. Given a solution x^*, the separation problem for constraints (36) calls for the determination of a subset S^* of nodes for which $\sum_{e\in\delta(S^*)} w_e x_e^*$ is a minimum: if such value is smaller than $\overline{W}$, a violated constraint is found and the process is iterated. Otherwise x^* is the optimal solution of the LP relaxation of the current subproblem. Thus, the separation problem amounts to finding a minimum-capacity cut in an undirected graph with edge capacities $w_e x_e^*$, and can be solved in polynomial time through max-flow techniques—very much in the spirit of separation of subtour elimination constraints for the TSP; see, e.g., Crowder and Padberg [13].

4.1 Model strengthening: a restart strategy

One of the most effective ingredients in modern ILP solvers is preprocessing and cut generation at the root node. However, its applicability to models where the initial formulation is not complete is problematic because the solver has only a limited view of the entire set of constraints. In order to avoid such troubles, we implemented a restart procedure that iteratively solves ILP model $M1_G$ (35)–(37) from scratch, each time enhancing the formulation. In particular, our restart approach for solving uncertain graph connectivity instances is as follows:

Restart
begin
1. compute an initial feasible solution using heuristics;
2. define an initial ILP relaxed $M1_G$ model as (32)–(34);
3. **repeat**
4. solve the current ILP model with a short time limit, separating and storing cuts (36) and possibly updating the incumbent;
5. add all stored cuts to the current ILP model;
6. **until** some halting condition is met;
7. solve the resulting model with the remaining time limit, possibly updating the incumbent.

end.

The same heuristic algorithm of Sect. 3.2.1 is used in Step 1 to provide initial heuristic solutions for the uncertain graph connectivity problem. At each execution of Step 4, we solve only the root node of the current formulation, so as to collect some relevant cuts in a reasonable amount of time. Note that the aim of this step is not to necessarily obtain an optimal solution of the problem, but to provide effective cuts that can be used in the subsequent iterations, and possibly to improve the current upper bound. The repeat-until loop is executed at most 9 times (or until the time limit is reached), so that the last execution of the solver (Step 7) works with a formulation where most of the "relevant" connectivity constraints (36) are included. When such constraints are included in the formulation and can be handled by preprocessing, the MIP solver `IBM ILOG Cplex` is usually able to produce in short time tight lower bounds on the optimal solution value, possibly proving the optimality of the current incumbent solution.

Restarting the solution of a MIP model is not a new idea in the literature. This technique is in fact very popular in the Constraint Programming community since the seminal papers by Luby et al. [28] and Gomes et al. [21]. Indeed, restart is routinely used within modern SAT solvers; see, e.g., Moskewicz et al. [32], Eén and Sörensson [14] and Goldberg and Novikov [19]. Recent advances on restarting techniques have been proposed by Williams et al. [38], Gomes [20] and Gomes and Walsh [22]. Restart can also be effective for solving hard MIP instances by using, e.g., the SCIP solver [1,34]. Recently, Karzan et al. [27] exploited a restart strategy where information on fathomed nodes before restart are collected so as to derive a "good" strategy for branching variable selection. Our perspective here is however quite different in that restart is intended to strengthen the formulation because a full model is not available at the beginning. This is also the reason why a similar restart technique did not produce a considerable speedup when applied to the USCP instances of Sect. 3.2.

4.2 Computational experiments

In our computational experiments, we considered all the TSPLIB [37] symmetric instances with at most 50 nodes. In order to simulate practical networks, for each instance defined by graph $\overline{G} = (V, \overline{E})$ and edge costs $\overline{c}_e$, we defined a sparse graph $G = (V, E)$ whose edge set is defined as follows. We start with $E = \emptyset$, compute a minimum cost spanning tree T of $\overline{G}$, and set $E = E \cup T$ and $\overline{E} = \overline{E} \backslash T$. This procedure is executed (at most) 5 times.

Each selected edge $e \in E$ has cost $c_e = \overline{c}_e$. Probabilities p_j associated with edges were randomly generated as in Sect. 3.2.2, i.e., according to a uniform distribution in [0, 0.2], and were used to determine coefficients w_j and $\overline{W}$ in the same way.

Tables 6, 7, 8 and 9 give, for each instance and for each value of $P^{\min} \in \{0.85, 0.90, 0.95, 0.99\}$, the following information:

- best solution found (z_h) by our initial heuristic—the associated computing times are omitted because they are always negligible,
- best solution found (z), best lower bound (LB), ratio z/LB, and computing time (T) for model $\mathrm{M1_G}$ defined by (35)–(37),
- best solution found (z), best lower bound (LB), ratio z/LB, and computing time (T) for model $\mathrm{M1'_G}$, in which the globally-valid constraints (24) with $k = 2$ and $k = 3$ are separated, at each node, by solving two additional min-cut problems with modified edge capacities. As to constraints (36), for numerical reasons we prefer not to generate them, unless this is required for the correctness of the method because the solution to be separated happens to be integer.
- best solution found (z), best lower bound (LB), ratio z/LB, and computing time (T) for the restart procedure described in Sect. 4.1 embedding model $\mathrm{M1'_G}$.

Table 6 Results on uncertain graph connectivity for $P^{\mathtt{min}} = 0.85$

Problem				Model $M1_G$				Model $M1'_G$				Model $M1'_G$+Restart			
Name	n	m	z_h	z	LB	z/LB	T	z	LB	z/LB	T	z	LB	z/LB	T
att48	48	235	9,696	9,696	6,706	1.45	T.L.	9,696	7,525	1.29	T.L.	9,696	8,187	1.18	T.L.
bayg29	29	140	1,373	1,373	1,058	1.30	T.L.	1,373	1,104	1.24	T.L.	1,373	1,172	1.17	T.L.
bays29	29	140	1,728	1,728	1,319	1.31	T.L.	1,728	1,473	1.17	T.L.	1,728	1,513	1.14	T.L.
burma14	14	65	2,671	2,671	2,671	1.00	42.31	2,671	2,671	1.00	0.67	2,671	2,671	1.00	0.83
dantzig42	42	205	642	642	480	1.34	T.L.	642	510	1.26	T.L.	642	538	1.19	T.L.
fri26	26	125	802	802	691	1.16	T.L.	787	768	1.03	T.L.	780	780	1.00	786.65
gr17	17	80	1,496	1,496	1,496	1.00	0.94	1,496	1,496	1.00	0.17	1,496	1,496	1.00	0.16
gr21	21	100	2,330	2,330	2,330	1.00	209.64	2,330	2,330	1.00	2.47	2,330	2,330	1.00	1.92
gr24	24	115	1,124	1,124	948	1.19	T.L.	1,124	1,065	1.06	T.L.	1,124	1,124	1.00	1,602.42
gr48	48	235	4,365	4,365	3,128	1.40	T.L.	4,365	3,405	1.28	T.L.	4,365	3,658	1.19	T.L.
hk48	48	235	11,013	11,013	7,387	1.49	T.L.	11,013	8,483	1.30	T.L.	11,013	8,889	1.24	T.L.
swiss42	42	205	1,199	1,199	840	1.43	T.L.	1,199	928	1.29	T.L.	1,199	994	1.21	T.L.
ulysses16	16	75	4,985	4,985	4,985	1.00	86.03	4,985	4,985	1.00	1.09	4,985	4,985	1.00	0.83
ulysses22	22	105	4,952	4,952	4,952	1.00	155.56	4,952	4,952	1.00	8.20	4,952	4,952	1.00	5.09
Arith. mean						1.22	1,192.46			1.14	1,158.04			1.09	1,071.28
# opt.							5				5				7

Table 7 Results on uncertain graph connectivity for $P^{\mathtt{min}} = 0.90$

Problem				Model $M1_G$				Model $M1'_G$				Model $M1'_G$+Restart			
Name	n	m	z_h	z	LB	z/LB	T	z	LB	z/LB	T	z	LB	z/LB	T
att48	48	235	10,997	10,997	7,013	1.57	T.L.	10,442	8,449	1.24	T.L.	10,366	8,997	1.15	T.L.
bayg29	29	140	1,566	1,566	1,097	1.43	T.L.	1,489	1,335	1.12	T.L.	1,465	1,382	1.06	T.L.
bays29	29	140	1,899	1,899	1,449	1.31	T.L.	1,797	1,690	1.06	T.L.	1,768	1,768	1.00	1,313.58
burma14	14	65	2,850	2,811	2,811	1.00	35.07	2,811	2,811	1.00	0.33	2,811	2,811	1.00	0.46
dantzig42	42	205	729	729	494	1.47	T.L.	665	593	1.12	T.L.	671	602	1.11	T.L.
fri26	26	125	894	894	708	1.26	T.L.	841	841	1.00	361.58	841	841	1.00	149.89
gr17	17	80	1,693	1,599	1,599	1.00	13.04	1,599	1,599	1.00	0.13	1,599	1,599	1.00	0.27
gr21	21	100	2,701	2,701	2,393	1.13	T.L.	2,554	2,554	1.00	17.00	2,554	2,554	1.00	6.85
gr24	24	115	1,233	1,233	990	1.25	T.L.	1,142	1,142	1.00	34.51	1,142	1,142	1.00	25.87
gr48	48	235	5,346	5,346	3,325	1.61	T.L.	4,661	3,972	1.17	T.L.	4,553	4,292	1.06	T.L.
hk48	48	235	12,910	12,910	8,016	1.61	T.L.	11,238	9,438	1.19	T.L.	10,991	10,308	1.07	T.L.
swiss42	42	205	1,349	1,349	900	1.50	T.L.	1,236	1,049	1.18	T.L.	1,212	1,113	1.09	T.L.
ulysses16	16	75	5,544	5,540	5,540	1.00	691.30	5,540	5,540	1.00	4.30	5,540	5,540	1.00	2.74
ulysses22	22	105	5,557	5,557	5,234	1.06	T.L.	5,410	5,410	1.00	6.62	5,410	5,410	1.00	4.65
Arith. mean						1.30	1,467.10			1.08	930.32			1.04	878.88
# opt.							3				7				8

Table 8 Results on uncertain graph connectivity for $P^{\mathtt{min}} = 0.95$

Problem				Model $M1_G$				Model $M1'_G$				Model $M1'_G$+Restart			
Name	n	m	z_h	z	LB	z/LB	T	z	LB	z/LB	T	z	LB	z/LB	T
att48	48	235	12,332	11,389	8,040	1.42	T.L.	10,258	9,868	1.04	T.L.	10,244	10,244	1.00	706.12
bayg29	29	140	1,905	1,751	1,316	1.33	T.L.	1,570	1,570	1.00	36.78	1,570	1,570	1.00	13.33
bays29	29	140	2,391	2,099	1,652	1.27	T.L.	1,945	1,945	1.00	3.79	1,945	1,945	1.00	1.66
burma14	14	65	4,039	3,182	3,182	1.00	21.16	3,182	3,182	1.00	0.08	3,182	3,182	1.00	0.21
dantzig42	42	205	911	754	573	1.32	T.L.	675	675	1.00	72.14	675	675	1.00	6.44
fri26	26	125	1,180	887	844	1.05	T.L.	874	874	1.00	1.52	874	874	1.00	1.24
gr17	17	80	2,197	1,807	1,807	1.00	3.15	1,807	1,807	1.00	0.08	1,807	1,807	1.00	0.21
gr21	21	100	3,258	2,584	2,584	1.00	95.28	2,584	2,584	1.00	0.19	2,584	2,584	1.00	0.66
gr24	24	115	1,503	1,288	1,142	1.13	T.L.	1,255	1,255	1.00	2.68	1,255	1,255	1.00	1.96
gr48	48	235	5,410	4,987	3,736	1.33	T.L.	4,528	4,528	1.00	699.93	4,528	4,528	1.00	47.46
hk48	48	235	15,754	12,841	8,953	1.43	T.L.	11,114	11,114	1.00	237.74	11,114	11,114	1.00	15.64
swiss42	42	205	1,652	1,474	1,036	1.42	T.L.	1,237	1,237	1.00	13.50	1,237	1,237	1.00	10.13
ulysses16	16	75	7,676	6,414	6,414	1.00	4.36	6,414	6,414	1.00	0.15	6,414	6,414	1.00	0.26
ulysses22	22	105	6,732	6,244	5,765	1.08	T.L.	5,999	5,999	1.00	0.20	5,999	5,999	1.00	0.49
Arith. mean						1.20	1,294.57			1.00	204.91			1.00	57.56
# opt.							4				13				14

Table 9 Results on uncertain graph connectivity for $P^{\mathtt{min}} = 0.99$

Problem				Model $M1_G$				Model $M1'_G$				Model $M1'_G$+Restart			
Name	n	m	z_h	z	LB	z/LB	T	z	LB	z/LB	T	z	LB	z/LB	T
att48	48	235	16,740	15,483	10,844	1.43	T.L.	13,946	11,685	1.19	T.L.	13,425	12,376	1.08	T.L.
bayg29	29	140	2,569	2,070	1,704	1.22	T.L.	1,875	1,875	1.00	428.42	1,875	1,875	1.00	17.72
bays29	29	140	3,281	2,631	2,222	1.18	T.L.	2,465	2,285	1.08	T.L.	2,426	2,426	1.00	19.74
burma14	14	65	4,859	3,881	3,881	1.00	2.20	3,881	3,881	1.00	1.11	3,881	3,881	1.00	0.56
dantzig42	42	205	1,134	948	738	1.28	T.L.	879	782	1.12	T.L.	831	831	1.00	1,291.79
fri26	26	125	1,445	1,081	1,040	1.04	T.L.	1,080	1,080	1.00	139.63	1,080	1,080	1.00	2.67
gr17	17	80	2,943	2,495	2,397	1.04	T.L.	2,466	2,466	1.00	277.45	2,466	2,466	1.00	1.12
gr21	21	100	3,716	3,113	3,113	1.00	101.62	3,113	3,113	1.00	13.41	3,113	3,113	1.00	1.32
gr24	24	115	2,128	1,724	1,444	1.19	T.L.	1,598	1,570	1.02	T.L.	1,592	1,592	1.00	6.77
gr48	48	235	7,618	7,618	4,867	1.57	T.L.	6,516	5,878	1.11	T.L.	6,338	6,064	1.05	T.L.
hk48	48	235	20,190	17,575	11,763	1.49	T.L.	15,963	13,395	1.19	T.L.	15,493	14,298	1.08	T.L.
swiss42	42	205	2,116	1,896	1,334	1.42	T.L.	1,676	1,444	1.16	T.L.	1,584	1,524	1.04	T.L.
ulysses16	16	75	9,766	7,770	7,770	1.00	8.50	7,770	7,770	1.00	0.38	7,770	7,770	1.00	0.80
ulysses22	22	105	9,621	7,061	7,061	1.00	1,359.48	7,061	7,061	1.00	55.23	7,061	7,061	1.00	1.43
Arith. mean						1.20	1,390.84			1.06	965.40			1.02	610.28
# opt.							4				7				10

A time limit of 1,800 s was given to each run. The last row of each table reports, for each solution method, the average computing time, in seconds (arithmetic mean; for the unsolved instances, the time limit is considered), and the number of instances solved to proven optimality within the given time limit.

Computational results clearly show that separation of constraints (24) before separation of (36) plays a relevant role for this problem. Indeed, model $\mathrm{M1_G}$ is able to solve only 16 instances (out of 56) within the given time limit, whereas model $\mathrm{M1'_G}$ solves 32 instances to optimality. On average, $\mathrm{M1'_G}$ is about 40 % faster than $\mathrm{M1_G}$, with an average gap $(z - LB)/LB$ of about 7 % (whereas for $\mathrm{M1_G}$ the gap is about 23 %).

Results are even better for the restart procedure, which finds the optimal solution of 39 instances (including all the 32 instances solved by $\mathrm{M1'_G}$), with an average computing time of about 650 seconds (20 % faster than $\mathrm{M1'_G}$). In addition, model $\mathrm{M1'_G}$+Restart finds better solutions and lower bounds than models $\mathrm{M1_G}$ and $\mathrm{M1'_G}$, the average gap $(z - LB)/LB$ being about 4 %.

The improved performance of the restart procedure is not surprising, in that more and more information on the solution structure is added to the model after each restart, so the MIP solver can take full advantage of it since the beginning of the solution process.

As in the USCP case, the initial heuristic is very good for $P^{\mathtt{min}} = 0.85$, and is improved only once (for `fri26`) within the time limit. As a matter of fact, in 6 out of 14 cases the initial solution turns out to be optimal, as certified by $\mathrm{M1'_G}$+Restart. As expected, however, the heuristic performance becomes worse for larger values of $P^{\mathtt{min}}$. If we let LB^* denote the best lower bound available for an instance, the average error $(z_h - LB^*)/LB^*$ is 10, 12, 25, and 32 % for $P^{\mathtt{min}} = 0.85$, 0.90, 0.95, and 0.99, respectively.

5 Conclusions

In this paper we have computationally analyzed the use of cutting planes for the solution of (integer) linear programs where the exact value of some input data is not known in advance.

Building upon the well-known concept of robustness due to Bertsimas and Sim [8], we have first analyzed a cutting plane solution approach for uncertain (integer) linear programs, pointing out situations where this method has a practical performance significantly better than the Bertsimas–Sim one—up to 2 orders of magnitude for some LP instances.

We have then considered problems whose uncertainty domain involves yes-no decisions that cannot be modeled by continuous variables, and analyzed alternative mathematical formulations for these uncertain problems. In particular, we have introduced a version of the well-known set covering problem arising when each column has a positive probability of disappearing and each row must be covered with a given probability. For this problem we have proposed two alternative (noncompact and compact, respectively) ILP models, and have computationally analyzed their performances. We have also studied an uncertain version of the classical minimum-cost graph connectivity problem, arising when edge failures occur with a certain probability and connectivity has to be guaranteed with high reliability. We have established the complexity of the problem, analyzed ad-hoc heuristics and proposed a sound restart procedure whose effectiveness has been proved through extensive computational tests.

A lesson learned from our experiments is that the cutting plane approach performs quite well for linear programs, but a compact formulation is preferable for (mixed) integer linear programs because it allows for a better use of the rich arsenal of preprocessing/cut-generation/branching tools available in modern MIP solvers. For the cases where such a compact ILP formulation is not available, our restart solution strategy may be a valid option.

Acknowledgments This work was supported by MiUR (PRIN 2009 project "Nonlinear Methods for Mixed-Integer Linear Programs"), and by the *Progetto di Ateneo* on "Computational Integer Programming" of the University of Padova. Thanks are also due to the referees for their helpful comments and suggestions.

References

1. Achterberg, T.: Constraint Integer Programming. Ph.D. thesis, ZIB, Technische Universität Berlin (2007)
2. Balas, E.: A class of location, distribution and scheduling problems: modeling and solution methods. In: Proc. Chinese-US Symposium on System Analysis. Wiley, New York (1983)
3. Ben-Tal, A., Nemirovski, A.: Robust solutions to uncertain linear programs. Oper. Res. Lett. **25**, 1–13 (1999)
4. Ben-Tal, A., Nemirovski, A.: Robust solutions of linear programming problems contaminated with uncertain data. Math. Progr. **88**, 411–424 (2000)
5. Ben-Tal, A., Nemirovski, A.: Robust optimization—methodology and applications. Math. Progr. **92**, 453–480 (2002)
6. Beraldi, P., Ruszczyński, A.: The probabilistic set-covering problem. Oper. Res. **50**, 956–967 (2002)
7. Bertsimas, D., Sim, M.: Robust discrete optimization and network flows. Math. Progr. **98**, 49–71 (2003)
8. Bertsimas, D., Sim, M.: The price of robustness. Oper. Res. **52**, 35–53 (2004)
9. Caprara, A., Fischetti, M., Toth, P.: Algorithms for the set covering problem. Ann. Oper. Res. **98**, 353–371 (2000)
10. Caprara, A., Fischetti, M., Toth, P., Vigo, D., Guida, P.L.: Algorithms for railway crew management. Math. Progr. **79**, 125–141 (1997)
11. Ceria, S., Nobili, P., Sassano, A.: A lagrangian-based heuristic for large-scale set covering problems. Math. Progr. **81**, 215–228 (1998)
12. Chen, X., Sim, M., Sun, P.: A robust optimization perspective on stochastic programming. Oper. Res. **55**, 1058–1071 (2007)
13. Crowder, H., Padberg, M.W.: Solving large-scale symmetric travelling salesman problems to optimality. Manag. Sci. **26**, 495–509 (1980)
14. Eén, N., Sörensson, N.: An extensible SAT-solver. In International Conference on Theory and Applications of Satisfiability Testing (SAT), LNCS, vol. 6. Springer, Berlin (2003)
15. Fischetti, M., Martello, S. : A hybrid algorithm for finding the k-th smallest of n elements in $O(n)$ time. In: Simeone, B., Toth, P., Gallo, G., Maffioli, F., Pallottino, S. (eds.) FORTRAN Codes for Network Optimization. Annals of Operational Research, vol. 13, pp. 401–419. JC Baltzer AG, Basel (1988)
16. Fischetti, M., Monaci, M.: Light robustness. In: Ahuja, R.K., Moehring, R., Zaroliagis, C. (eds.) Robust and Online Large-Scale Optimization, Lecture Notes in Computer Science, vol. 5868, pp. 61–84. Springer, Berlin (2009)
17. Gamache, M., Soumis, F., Marquis, G., Desrosiers, J.: A column generation approach for large-scale aircrew rostering problems. Oper. Res. **47**, 247–263 (1999)
18. Garey, M., Johnson, D.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman, San Francisco (1979)
19. Goldberg, E., Novikov, Y.: Berkmin: a fast and robust sat-solver. Discrete Appl. Math. **155**, 1549–1561 (2007)

20. Gomes, C.: Complete randomized backtrack search. In: Milano, M. (ed.) Constraint and Integer Programming: Toward a Unified Methodology, pp. 233–283. Kluwer, Dordrecht (2003)
21. Gomes, C., Selman, B., Kautz, H.: Boosting combinatorial search through randomization. In: Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98). American Association for Artificial Intelligence (1998)
22. Gomes, C., Walsh, T.: Randomness and structure. In: Rossi, F., van Beek, P., Walsh, T. (eds.) Handbook of Constraint Programming, Elsevier, Amsterdam (2006)
23. Grötschel, M., Monma, C.L.: Integer polyhedra arising from certain network design problems with connectivity constraints. SIAM J. Discrete Math. **3**, 502–523 (1990)
24. Grötschel, M., Monma, C.L., Stoer, M.: Computational results with a cutting plane algorithm for designing communication networks with low-connectivity constraints. Oper. Res. **40**, 309–330 (1992)
25. Haight, R.G., Revelle, C., Snyder, S.A.: An integer optimization approach to a probabilistic reserve site selection problem. Oper. Res. **48**, 697–708 (2000)
26. IBM ILOG CPLEX Optimizer. http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/
27. Karzan, F.K., Nemhauser, G.L., Savelsbergh, M.W.P.: Information-based branching schemes for binary linear mixed integer problems. Math. Progr. Comput. **1**, 249–293 (2009)
28. Luby, M., Sinclair, A., Zuckerman, D.: Optimal speedup of las vegas algorithms. Inf. Process. Lett. **47**, 173–180 (1993)
29. Martello, S., Pisinger, D., Toth, P.: Dynamic programming and strong bounds for the 0-1 knapsack problem. Manag. Sci. **45**, 414–424 (1999)
30. Martello, S., Toth, P.: Knapsack Problems: Algorithms and Computer Implementations. Wiley, Chichester (1990)
31. Monma, C.L., Munson, B.S., Pulleyblank, W.R.: Minimum-weight two-connected spanning networks. Math. Progr. **46**, 153–171 (1990)
32. Moskewicz, M.W., Madigan, C.F., Zhao, Y., Zhang, L., Malik, S.: Chaff: engineering an efficient SAT solver. In: Proceedings of the 38th Annual Design Automation Conference, pp. 530–535. ACM, New York (2001)
33. ORLIB. http://people.brunel.ac.uk/~mastjjb/jeb/info.html
34. SCIP: Solving Constraint Integer Programs. http://scip.zib.de/
35. Saxena, A., Goyal, V., Lejeune, M.: MIP reformulations of the probabilistic set covering problem. Math. Progr. **121**, 1–31 (2010)
36. Soyster, A.L.: Convex programming with set-inclusive constraints and applications to inexact linear programming. Oper. Res. **21**, 1154–1157 (1973)
37. TSPLIB. http://comopt.ifi.uni-heidelberg.de/software/tsplib95/
38. Williams, R., Gomes, C.P., Selman, B.: Backdoors to typical case complexity. In: Gottlob, G., Walsh, T. (eds) IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, pp. 1173–1178. Morgan Kaufmann, Waltham (2003)