Valid inequalities for a robust knapsack polyhedron - Application to the robust bandwidth packing problem

Olivier Klopfenstein[1,2] Dritan Nace[2]

[1] *Orange Labs*
38-40 rue du gl Leclerc, 92130 Issy-les-Moulineaux, France

[2] *Université de Technologie de Compiègne, Laboratoire Heudiasyc UMR CNRS 6599*
60205 Compiègne Cedex, France

Abstract

Knapsack cover inequalities are among the most frequently used cuts for solving 0-1 linear programs. In this paper, we propose adaptations of these classical valid inequalities to the case when knapsack items have uncertain weights. For modeling uncertainty, we focus on the robust framework proposed by Bertsimas and Sim (2003). Then, a robust version of the bandwidth packing problem is considered, when traffic demands in a network are subject to incertainty. Numerical experiments show the dramatic improvement brought by using the proposed cover inequalities when solving this problem.

Keywords: *Robust optimization, knapsack problem, cover inequalities, bandwidth packing problem, routing*

1 Introduction

Many real-life problems require uncertainty on input data to be taken into account. Earlier work on robust optimization has provided interesting ways to handle uncertainty on input data for linear problems (see [2, 4]). The proposed models involve a parameter which enables to controle the level of robustness. The one of Bertsimas and Sim [4] has the advantage of preserving the linearity of the initial problem. Hence, the robust version of an integer linear problem is also an integer linear problem. This is important in practice, since integer linear solvers are nowadays very efficient. Despite its practical interest, the application of this general model of robustness to integer linear problems has not been extensively investigated so far. In the literature, the theoretical studies on applications of this framework for robust integer programming have mainly focused on cases where only objective coefficients are uncertain. Within this restriction, [3] has provided complexity results, [14] proves a probability bound easier to compute than that proposed in [4]. Some reformulations are investigated in [1] for this particular case of uncertain objective coefficients.

However, many uncertain problems also require uncertain constraint coefficients to be considered. With respect to the robust framework of Bertsimas and Sim, this uncertainty impacts the polyhedron of feasible solutions. Because of its fundamental and practical importance for integer programming, this paper addresses the 0-1 knapsack problem with uncertain weights and capacity. The stress is put on adaptating classical results to obtain strong polyhedral descriptions for robust problems. In particular, this paper shows how the widely used knapsack cover cuts can be adapted to provide strong inequalities for the robust knapsack problem. Then, these inequalities are applied to a robust unsplittable routing problem. Computational tests show the interest of considering the proposed inequalities for solving such combinatorial problems.

Due to paper size constraints, proofs have been omitted.

2 On the robust knapsack polyhedron

Let $I = \{1, \dots, n\}$. The classical knapsack polyhedron is: $\mathcal{K} = conv\left\{x \in \{0,1\}^n | \sum_{i \in I} w_i x_i \leq c\right\}$, where coefficients w_i (weights) and c (knapsack capacity) are assumed to be non-negative integers. Given a profit vector p, the knapsack problem is: $\max\{px | x \in \mathcal{K}\}$. For an extensive study of the knapsack problem and its variants, we refer to [10, 8]. The problem is known to be NP-hard, but there exists a pseudo-polynomial dynamic programming algorithm to solve it. While branch-and-cut algorithms are often not the best suited for solving this problem, its polyhedral comprehension is of interest. Indeed, knapsack appears as a sub-problem of a lot of other larger problems which are classically processed with branch-and-cut, since any single linear inequality with 0-1 variables can be equivalently written as a knapsack constraint.

From now on, we assume that each coefficient w_i lies in an interval $[\overline{w}_i - \hat{w}_i, \overline{w}_i + \hat{w}_i]$: $\overline{w}_i > 0$ is the nominal value of the weight w_i, $\hat{w}_i \geq 0$ being the possible variation of w_i from this expected value. Both $\overline{w}$ and $\hat{w}$ are assumed to be integral vectors. To each weight is associated a random variable; for the sake of simplicity, the random variables and their realizations are denoted by the same symbol w_i.

Let $\Gamma \in [0, n]$. The idea of the robust formulation is to find a solution which is feasible even though up to Γ coefficients of w (when Γ is integral) take their largest possible value. Thus, for instance, if $\Gamma = 0$, we consider only the nominal scenario (when w is supposed to take the value $\overline{w}$); if $\Gamma = n$, the worst case (weights $\overline{w} + \hat{w}$) is taken into account. Within this framework, let us introduce the robust knapsack polyhedron, parameterized by Γ:

$$\mathcal{K}(\Gamma) = conv\Big\{x \in \{0,1\}^n \;|\forall S \subseteq I \text{ s.t. } |S| = \lceil\Gamma\rceil : \sum_{i \in I \setminus S} \overline{w}_i x_i + \sum_{i \in S} (\overline{w}_i + \hat{w}_i) x_i - (\lceil\Gamma\rceil - \Gamma). \min_{i \in S}\{\hat{w}_i x_i\} \leq c\Big\}. \tag{1}$$

Considering Γ possibly fractional makes formulas a bit complex. For the sake of clarity, the following presentation is restricted to the case when Γ is integral: $\Gamma \in \{1, \dots, n\}$. Nevertheless, note that all the results can be directly adapted to the fractional case.

The following result holds:

Lemma 1 *Let $\Gamma \in \{1, \dots, n\}$. $\mathcal{K}(\Gamma)$ is full-dimensional if, and only if, for all $i \in I$, $\overline{w}_i + \hat{w}_i \leq c$.*

From now on, we assume that none of the robust weights $\overline{w}_i + \hat{w}_i$ exceeds the knapsack capacity. For any subset $S \subseteq I$, let us denote:

$$\mathcal{K}_S = \Big\{x \in \{0,1\}^n | \sum_{i \in S} (\overline{w}_i + \hat{w}_i) x_i + \sum_{i \in I \setminus S} \overline{w}_i x_i \leq c\Big\}. \tag{2}$$

Lemma 2 *Let $S \subseteq I$ such that $|S| \leq \Gamma$, any inequality valid for $\mathcal{K}_S$ is valid for $\mathcal{K}(\Gamma)$.*

This is clear, since $\mathcal{K}(\Gamma) = conv\left(\bigcap_{S \subseteq I, |S| = \Gamma} \mathcal{K}_S\right)$. However, most of the time: $\mathcal{K}(\Gamma) = conv\left(\bigcap_{S \subseteq I, |S| = \Gamma} \mathcal{K}_S\right) \subset \bigcap_{S \subseteq I, |S| = \Gamma} conv(\mathcal{K}_S)$.

Let us show now how classical results related to the knapsack polyhedron can be adapted for its robust version. When dealing with the classical knapsack problem, the most frequently used cutting inequalities are the so-called *knapsack cover inequalities* (or cover inequalities for short, see e.g. [11]). The classical concepts related to cover inequalities can be adapted to our robust framework:

Definition 1 *The set $\mathcal{C} \subseteq I$ is a robust cover if there exists $S \subseteq \mathcal{C}$ such that $|S| = \min\{\Gamma, |\mathcal{C}|\}$ and:* $\sum_{i \in \mathcal{C} \setminus S} \overline{w}_i + \sum_{i \in S} (\overline{w}_i + \hat{w}_i) > c$.

Definition 2 *A robust cover $\mathcal{C} \subseteq I$ is said minimal if for any $i \in \mathcal{C}$, $\mathcal{C} \setminus \{i\}$ is not a robust cover.*

For any robust cover $\mathcal{C}$, the following inequality is valid for $\mathcal{K}(\Gamma)$:

$$\sum_{i \in \mathcal{C}} x_i \leq |\mathcal{C}| - 1.$$

Such an inequality will be called a robust cover inequality (or cut). That this inequality is valid for $\mathcal{K}(\Gamma)$ is clear, since it is valid for at least one of the sets $\mathcal{K}_S$ (cf Lemma 2). On the other hand, the classical concept of *extended cover* can be adapted to our robust framework (note that extended cover inequalities are known to be very efficient in practice, see e.g. [5]). A robust cover $\mathcal{C}$ can be extended into $E(\mathcal{C})$:

$$E(\mathcal{C}) = \begin{cases} \mathcal{C} \cup \{i \notin \mathcal{C} \,|\, \overline{w}_i + \hat{w}_i \geq \max_{k \in \mathcal{C}} (\overline{w}_k + \hat{w}_k)\}, & \text{if } |\mathcal{C}| \leq \Gamma \\ \mathcal{C} \cup \{i \notin \mathcal{C} \,|\, \overline{w}_i \geq \max_{k \in \mathcal{C}} \overline{w}_k, \text{ and } \overline{w}_i + \hat{w}_i \geq \max_{k \in \mathcal{C}} (\overline{w}_k + \hat{w}_k)\}, & \text{if } |\mathcal{C}| \geq \Gamma + 1 \end{cases}. \quad (3)$$

As in the classical framework, $E(\mathcal{C})$ extends the cover $\mathcal{C}$ with elements of "weights not less than the largest weight in $\mathcal{C}$". In our robust context, this statement concerns only maximal weights of elements when $|\mathcal{C}| \leq \Gamma$. On the contrary, when $|\mathcal{C}| \geq \Gamma + 1$, both the nominal and the maximal weights of elements have to be considered.

Proposition 1 *Let $\mathcal{C}$ be a robust cover and $E(\mathcal{C})$ its extension, the following inequality is valid for $\mathcal{K}(\Gamma)$:*

$$\sum_{i \in E(\mathcal{C})} x_i \leq |\mathcal{C}| - 1. \quad (4)$$

While robust cover cuts are valid for some knapsack sets $\mathcal{K}_S$, this is not the case, in general, for extended robust cover inequalities. Simple examples can be built, for which extended robust cover cuts are valid for none of the sets $\mathcal{K}_S$.

Let us observe that, apart from providing stronger cuts for the robust problem, the theoretical setting proposed enables us to characterize them in a global way. That is, considering all the classical knapsack sets $\mathcal{K}_S$ is avoided and efficiently replaced by a more general analysis. This is of primary importance from both the theoretical and practical points of view.

Most of the classical polyhedral results available for cover cuts (see for instance [11]) can be transposed directly in our robust framework.

Proposition 2 *Suppose that the set I is ordered so that: $i < j \Rightarrow \begin{cases} \overline{w}_i \geq \overline{w}_j \\ \overline{w}_i + \hat{w}_i \geq \overline{w}_j + \hat{w}_j \end{cases}$. Let $\mathcal{C} = \{i_1, ..., i_r\}$ be a minimal robust cover, with $i_1 < i_2 < ... < i_r$. If any of the following conditions holds, then (4) is a facet of $\mathcal{K}(\Gamma)$:*
(i) $\mathcal{C} = I$,
(ii) $E(\mathcal{C}) = I$, and $(\mathcal{C} \setminus \{i_1, i_2\}) \cup \{1\}$ is not a robust cover,
(iii) $E(\mathcal{C}) = \mathcal{C}$, and $(\mathcal{C} \setminus \{i_1\}) \cup \{p\}$ is not a robust cover, where $p = \min\{i \in I \setminus E(\mathcal{C})\}$,
(iv) $\mathcal{C} \subset E(\mathcal{C}) \subset I$, and neither $(\mathcal{C} \setminus \{i_1, i_2\}) \cup \{1\}$ nor $(\mathcal{C} \setminus \{i_1\}) \cup \{p\}$, with $p = \min\{i \in I \setminus E(\mathcal{C})\}$, are robust covers.

The ordering condition on weights is theoretically restrictive, while corresponding to many applicative frameworks. Other results can be derived without this hypothesis.

Considering the classical knapsack polyhedron, the separation of cover inequalities is known also to be a knapsack problem. This separation problem can be adapted for the robust knapsack polyhedron $\mathcal{K}(\Gamma)$. We prove that a most violated robust cover inequality can be generated by solving:

$$\begin{array}{ll} \min & \sum_{i \in I} (1 - \tilde{x}_i) r_i \\ \text{s.c.} & \sum_{i \in I} \left(\overline{w}_i r_i + \hat{w}_i s_i\right) \geq c + 1, \\ & \sum_{i \in I} s_i \leq \Gamma, \\ & s_i \leq r_i, \qquad \forall i \in I, \\ & r \in \{0,1\}^n, s \in [0,1]^n. \end{array} \quad (5)$$

The optimal value of (5) is (strictly) less than 1 if and only if a violated robust cover inequality exists. In this case, the solution vector r characterizes a violated cover. As for classical separation of cover cuts:

Lemma 3 *If for all $i \in I$, $\tilde{x}_i < 1$, then the robust cover produced by (5) is minimal.*

This separation problem is NP-hard, since $\Gamma = 0$ leads to a knapsack problem. For practical solution, we propose a natural adaptation of the well known greedy heuristic used to solve approximately classical knapsack problems (cf [10]) (not detailed in this article).

3 Solving the robust bandwidth packing problem

The Bandwidth Packing Problem (BPP) consists of maximizing the profit associated to a multicommodity flow in a network, while routing each flow on a single path. Let V denotes the set of network vertices, A the set of directed arcs; $c_a > 0$ is the capacity of an arc $a \in A$. For each $v \in V$, $A^-(v)$ is the set of incoming arcs at node v; similarly, $A^+(v)$ is the set of outgoing arcs at node v. Let us denote by I the set of demands: each demand $i \in I$ is characterized by a source node $s_i \in V$, a destination node $t_i \in V$, a bandwidth amount $b_i > 0$ and an associated profit $p_i > 0$. For each $(i, a) \in I \times A$, the variable x_{ia} takes value 1 when the demand i is routed on arc a; otherwise, $x_{ia} = 0$.

Then the Bandwidth Packing Problem can be written as:

$$\begin{array}{lll}
\max & \sum_{i\in I} p_i \left(\sum_{a\in A^+(s_i)} x_{ia}\right) & \\
\text{s.t.} & \sum_{a\in A^+(v)} x_{ia} - \sum_{a\in A^-(v)} x_{ia} = 0, & \forall i \in I, \forall v \in V \setminus \{s_i, t_i\}, \quad (6)\\
& \sum_{a\in A^+(s_i)} x_{ia} \leq 1, & \forall i \in I, \quad (7)\\
& \sum_{i\in I} b_i x_{ia} \leq c_a, & \forall a \in A, \quad (8)\\
& x_{ia} \in \{0, 1\}, & \forall (i, a) \in I \times A.
\end{array}$$

Note that $\sum_{a\in A^+(s_i)} x_{ia} = 1$ when the demand i is routed. Constraints (6) correspond to flow conservation in the network. Equations (7) give the traffic condition at the source node (similar conditions on the destination node are unnecessary, since they are implied by (6) and (7)). Equations (8) are the capacity constraints.

This classical problem has been widely studied, and cover cuts have been successfully used to improve the solution process (see e.g. [13]). This is the motivation for testing if the proposed robust cut inequalities have the same impact in a robust context. Let us suppose now that bandwidth demands are uncertain: $b_i \in [\overline{b}_i - \hat{b}_i, \overline{b}_i + \hat{b}_i]$ for each $i \in I$, with $\hat{b}_i \geq 0$ and $\overline{b}_i \geq \hat{b}_i$. $\overline{b}_i$ is the nominal bandwidth value expected for demand i. Using the robust approach of Bertsimas and Sim, let us introduce a robust parameter $\Gamma \in [0, n]$, with $n = |I|$. When Γ is integral, a routing x will be a solution of the Robust Bandwidth Packing Problem (RBPP) when at most Γ traffic demands can take their largest bandwidth values, while the $n - \Gamma$ others take their nominal values. The corresponding mathematical formulation is given below:

$$\begin{array}{ll}
\max & \sum_{i\in I} p_i \left(\sum_{a\in A^+(s_i)} x_{ia}\right) \\
\text{s.t.} & \forall S \subseteq I, |S| = \Gamma : \left\{ \begin{array}{ll}
\sum_{a\in A^+(v)} x_{ia} - \sum_{a\in A^-(v)} x_{ia} = 0, & \forall i \in I, \forall v \in V \setminus \{s_i, t_i\}, \\
\sum_{a\in A^+(s_i)} x_{ia} \leq 1, & \forall i \in I, \\
\sum_{i\in S} (\overline{b}_i + \hat{b}_i) x_{ia} + \sum_{i\notin S} \overline{b}_i x_{ia} \leq c_a, & \forall a \in A, \\
x_{ia} \in \{0, 1\}, & \forall (i, a) \in I \times A.
\end{array}\right.
\end{array} \quad (9)$$

Note that this robust model is easily extended to the case when Γ is fractional; the presentation is kept with Γ integral for the sake of simplicity. Related robust routing problems have already been introduced, mostly with splittable flows, see e.g. [3, 12]. The work [3] focuses on uncertainties in costs only (objective function only). By contrast, [12] considers uncertainty on demand bandwidth and looks for an optimal robust network dimensioning. With respect to unsplittable routing, note that [3] introduces the robust shortest path problem.

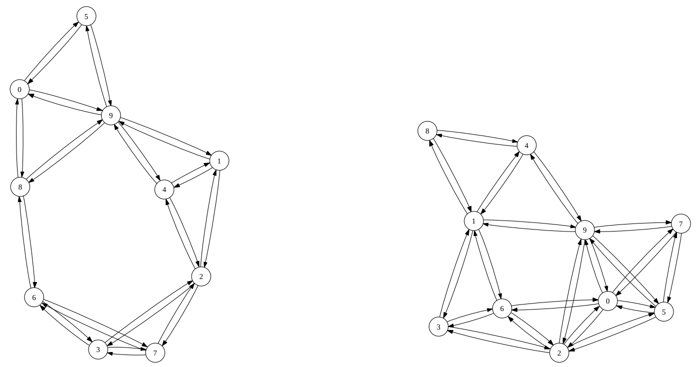

Figure 1: Two 10-nodes network topologies.

In the above model, robustness impacts only on capacity constraints. The above formulation can equivalently be replaced with the following one:

$$\begin{array}{lll} \max & \sum_{i\in I} p_i \left(\sum_{a\in A^+(s_i)} x_{ia}\right) & \\ \text{s.t.} & \sum_{a\in A^+(v)} x_{ia} - \sum_{a\in A^-(v)} x_{ia} = 0, & \forall i \in I, \forall v \in V \setminus \{s_i, t_i\}, \\ & \sum_{a\in A^+(s_i)} x_{ia} \leq 1, & \forall i \in I, \\ & \sum_{i\in S}(\overline{b}_i + \hat{b}_i)x_{ia} + \sum_{i\notin S} \overline{b}_i x_{ia} \leq c_a, & \forall a \in A, \forall S \subseteq I, |S| = \Gamma, \\ & x_{ia} \in \{0,1\}, & \forall (i,a) \in I \times A. \end{array} \tag{10}$$

This latter formulation outlines more precisely the impact of parameter Γ. Following the general model, Γ could take values up to n. But in our specific application, we see that what is important is not the global number of demands affected by robustness, but the number of demands *per link*. As a result, it is unnecessary to consider Γ larger than the number of demands using a given link. This is more precisely expressed in the next result:

Lemma 4 *If:* $\Gamma \geq \dfrac{\max_{a\in A} c_a}{\min_{i\in I}(\overline{b}_i + \hat{b}_i)}$, *then the solution of RBPP corresponds to the worst case, when all traffic demands take their largest values.*

Note that this simple condition on Γ is not tight, and can be refined.

Our aim is to illustrate how efficient robust cover cuts can be in practice. Numerical tests have been performed on two bidirected network topologies with 10 nodes, see Figure 1. All arcs a have the same capacity $c(a) = 100$. 40 demands, i.e. triplets $(s_i, t_i, \overline{b}_i)$, are randomly generated; the nominal bandwidth values $\overline{b}_i$ are integral, between 20 and 60. The bandwidth is subject to uncertainty: for each demand i, $\hat{b}_i = \alpha.\overline{b}_i$, where $\alpha \in (0, 1]$. Finally, the profits p_i are integers randomly chosen between 10 and 20.

All the tests rely on Cplex, and solve a polynomial-size formulation of RBPP directly derived from the results of [4] (not detailed here). Four solution strategies are compared. The first one, denoted by "Cplex" in the result tables, consists of running Cplex in default mode on the model. The second strategy is the branch-and-bound algorithm of Cplex, with no cut generation; it is denoted by "B&B". The third approach, denoted by "r.c.c.", is to use the branch-and-bound algorithm of Cplex (without Cplex automatic cut generation), while generating robust cover cuts. The fourth strategy is similar, except we generate extended robust cover cuts; it is denoted by "extended r.c.c.".

When generating our own cover cuts, the heuristic separation algorithm mentioned at the end of Section 2 is used. When visiting a new branching node, the current linear program is solved, and a (fractional) solution is obtained; then, all the violated inequalities found are added to the model, and the updated linear program is solved again before going to another branching node. Hence, only one cut generation round is performed at each branching node.

network	Γ	Cplex		B&B		r.c.c.			extended r.c.c.			value
		NN	T	NN	T	NC	NN	T	NC	NN	T	
1	0	201	2	806653	>1000 (0.80%)	2219	80756	>1000 (1.91%)	828	7057	77	483
1	1	41390	>1000 (4.18%)	131687	>1000 (5.96%)	3322	23724	>1000 (1.67%)	1291	9223	299	443
1	2	45469	>1000 (6.53%)	127301	>1000 (10.18%)	3005	25607	>1000 (1.27%)	735	3082	88	423
1	3	40679	>1000 (6.93%)	138647	>1000 (6.19%)	2387	23717	>1000 (0.68%)	584	1318	54	420
2	0	0	1	4990	11	567	1843	72	767	1448	38	559
2	1	13795	>1000 (10.81%)	91011	>1000 (10.46%)	3923	7787	>1000 (6.11%)	567	288	57	515
2	2	19899	>1000 (24.41%)	71700	>1000 (11.79%)	4351	7924	>1000 (16.15%)	1580	7390	>1000 (0.48%)	≥491
2	3	11479	>1000 (25.53%)	54401	>1000 (16.21%)	3763	8731	>1000 (13.94%)	1154	2845	359	490

Table 1: Numerical results with 40 demands and $\alpha = 30\%$.

A total solution time limit is fixed at 1000 seconds; when this limit is reached, the proved gap to optimum is indicated. NN denotes the number of branching nodes visited; T is the total solution time in seconds (CPU time). When using robust cover cuts, NC is the total number of cuts added in the branching tree.

Given the way instances are defined, and from the observation of Lemma 4, we deduce that it is useless to consider $\Gamma \geq 4$. Indeed, all arcs have capacity 100, while all demands have a nominal value at least 20; since uncertainty is 30% of the nominal value, the variation of a demand bandwidth is at least 6. Observing that $100/(20+6) \simeq 3.8$, $\Gamma = 4$ would already correspond to the worst case, when all traffic demands take their largest values.

Results appear in Table 1. The lines with $\Gamma = 0$ correspond to classical BPP instances, which are solved very efficiently by Cplex. In this usual context, the techniques of this powerful software are far more efficient than just adding cover inequalities as we do. In fact, even a simple branch-and-bound algorithm may prove more efficient than our approaches. However, when looking at robust contexts ($\Gamma > 0$), our specialized strategy appears a lot more efficient than what Cplex can do, at least on the highly combinatorial instances we have considered. We first observe that, even though simple robust cover inequalities may have some interest, extended inequalities are much more efficient. When using extended robust cover cuts, we observe that all the considered instances (except one) are solved in relatively short time by our approach, while Cplex reveals unable to prove optimality for most of the instances within the time limit (1000 seconds). Note that proved gaps to optimum from Cplex may remain very large, up to 25% (see network 2, $\Gamma = 3$).

The only exception is for network 2 with $\Gamma = 2$, for which the proposed approach fails to prove optimality within the time limit of 1000 seconds. However, observe that the given gap is small (0.48%). For this specific instance, 1381 seconds were necessary for our method to prove optimality.

References

[1] A. Atamtürk, Strong formulations of robust mixed 0-1 programming, submitted to Math. Prog., available at www.optimization-online.org/DB_FILE/2004/01/816.pdf (2004).

[2] A. Ben-Tal and A. Nemirovski, Robust solutions of Linear Programming problems contamined with uncertain data, Math. Program. (Ser. A), 88 (2000), pp. 411-424.

[3] D. Bertsimas and M. Sim, Robust discrete optimization and network flows, Math. Program. (Ser. B), 98 (2003), pp. 49-71.

[4] D. Bertsimas and M. Sim, The Price of Robustness, Operations Research, 52-1 (2004), pp. 35-53.

[5] V. Gabrel and M. Minoux, A scheme for exact separation of extended cover inequalities and application to multidimensional knapsack problems, Operations Research Letters, 30 (2002), pp. 252-264.

[6] Z. Gu, G.L. Nemhauser and M.W.P. Savelsbergh, Lifted cover inequalities for 0-1 integer programs: computation, INFORMS J. Comput., 10 (1998), pp. 427-437.

[7] Z. Gu, G.L. Nemhauser and M.W.P. Savelsbergh, Lifted cover inequalities for 0-1 integer programs: complexity, INFORMS J. Comput., 11-1 (1999), pp. 117-123.

[8] H. Kellerer, U. Pferschy and D. Pisinger, Knapsack Problems, Springer (2004).

[9] O. Klopfenstein and D. Nace, A robust approach to the chance-constrained knapsack problem, Operations Research Letters, 36 (2008), pp. 628-632.

[10] S. Martello and P. Toth, Knapsack Problems: Algorithms and Computer Implementations, Wiley (1990).

[11] G.L. Nemhauser and L.A. Wolsey, Integer and Combinatorial Optimization, John Wiley & Sons (1999).

[12] A. Ouorou and J.-Ph. Vial, A model for robust capacity planning for telecommunications networks under demand uncertainty, *Proc. of the workshop on Design of Reliable Communication Networks (DRCN)*, 2007.

[13] K. Park, S. Kang and S. Park, An integer programming approach to the bandwidth packing problem, *Management Science*, Vol. 42, No. 9 (1996), pp. 1277-1291.

[14] M.C. Pinar, A Note on Robust 0-1 Optimization with Uncertain Cost Coefficients, 4OR, Quarterly Journal of the Belgian, French and Italian Operations Research Societies, Vol. 2 (2004), pp. 309-316.