Robust combinatorial optimization problems under locally budgeted interdiction uncertainty against the objective function and covering constraints

Alejandro Crema
Universidad Central de Venezuela
alejandro.crema@ciens.ucv.ve

Fernando Crema García
KU Leuven
fernando.cremagarcia@esat.kuleuven.be

November 24, 2025

Abstract

Recently robust combinatorial optimization problems with budgeted interdiction uncertainty affecting cardinality-based constraints or objective were considered by presenting, comparing and experimenting with compact formulations. In this paper we present a compact formulation for the case in which locally budgeted interdiction uncertainty affects the objective function and covering constraints simultaneously.

1 Introduction

Let $m, n \geq 1$, let $A \in \{0,1\}^{m \times n}$ be the *covering constraints matrix*, let $\mathbf{q} \in \mathbb{R}_+^n$ the *profit vector* and let $\mathcal{X} \subseteq \{0,1\}^n$. Let us suppose that $\mathcal{X}$ is 0-1-Mixed Integer Linear Programming (0-1-MILP) representable. If $m \in \mathbb{N}$ let $[m] = \{1, \cdots, m\}$ and let $[0] = \emptyset$. The *nominal problem* to be considered in this paper is a problem in $\boldsymbol{x}$ defined as follows:

$$
\begin{aligned}
\max \; & \sum_{j \in [n]} \mathbf{q}_j \boldsymbol{x}_j & & \mathcal{P} \\
s.t. \; & \sum_{j \in [n]} A_{ij} \boldsymbol{x}_j \geq 1 & \forall i \in [m] & \\
& \boldsymbol{x} \in \mathcal{X}
\end{aligned}
$$

The nominal problem presented has a very general structure and several problems may be interpreted as $\mathcal{P}$. Some examples are: the *combined knapsack-set covering problem* (see [1]) and variants of the *multiple-choice knapsack problem* (see [2]).

In [3] the *robust combinatorial optimization problem with budgeted interdiction uncertainty* affecting *cardinality-based constraints* (which include covering constraints) or *objective* was considered and several compact formulations were presenting and comparing from theorethical and practical point of view. *Locally budgeted uncertainty* was presented in [4] to model uncertainty against the objective function with uncertain parameters partitioned such that a classic budgeted uncertainty set applies to each part of the partition. In this paper we study the case in which *locally budgeted interdiction uncertainty* affects the objective function and covering constraints simultaneously.

Let $R \geq 1$, let $\{J_r\}_1^R$ be a partition of $[n]$ with $|J_r| = n_r \geq 1$ for all $r \in [R]$, let $J_r^1 \subseteq J_r$ for all $r \in [R]$, let $\mathbf{\Gamma} \in \mathbb{N}_+^R$ and let $\mathcal{U}$ be the *locally budgeted interdiction uncertainty set* implicitly defined as follows:

$$\mathcal{U} = \left\{ \boldsymbol{s} \in \{0,1\}^n : \sum_{j \in J_r^1} \boldsymbol{s}_j \leq \mathbf{\Gamma}_r \ \forall r \in [R],\ \boldsymbol{s}_j = 0 \ \forall j \in J_r - J_r^1 \ \forall r \in [R] \right\}$$

Let $NV_r = \{j \in J_r - J_r^1\}$ for all $r \in [R]$ and let $\mathcal{NV} = \cup_{r \in [R]} NV_r$. If $j \in \mathcal{NV}$ we have that $\boldsymbol{s}_j = 0$ for all $\boldsymbol{s} \in \mathcal{U}$ and $\boldsymbol{x}_j$ is a *nonvulnerable* variable. Let $W \in \mathbb{N}$ with $W \geq \mathbf{\Gamma}_r + 1$ for all $r \in [R]$ and let $\mathbf{w} \in \{1, W\}^n$ defined as follows: $\mathbf{w}_j = \begin{cases} 1 & \text{if} \quad j \in [n] - \mathcal{NV} \\ W & \text{if} \quad j \in \mathcal{NV} \end{cases}$ for all $j \in [n]$ then $\mathcal{U}$ may be rewritten as follows:

$$\mathcal{U} = \left\{ \boldsymbol{s} \in \{0,1\}^n : \sum_{j \in J_r} \mathbf{w}_j \boldsymbol{s}_j \leq \mathbf{\Gamma}_r \ \forall r \in [R] \right\}$$

Let $\boldsymbol{x} \in \{0,1\}^n$ then the *worst scenario problem for* $\boldsymbol{x}$ is a problem in $\boldsymbol{s}$ defined as follows:

$$\min \sum_{j \in [n]} \mathbf{q}_j \boldsymbol{x}_j \left(1 - \boldsymbol{s}_j\right) \ \ s.t. \ \ \boldsymbol{s} \in \mathcal{U} \qquad \mathcal{WS}(\boldsymbol{x})$$

Hereafter: (i) if $\bullet$ is an optimization problem then $F(\bullet)$ is its feasible solution set and $v(\bullet)$ is its optimal value (if it exists) and (ii) the vector with all elements equal to 0(1) will be denoted $\mathbf{0}(\mathbf{1})$ and the dimension will be clear in the context.

The *Robust combinatorial optimization problem under locally budgeted interdiction uncertainty against the objective function and covering constraints* is a problem in $\boldsymbol{x}$ defined as follows:

$$\begin{aligned} \max \ & v\Big(\mathcal{WS}(\boldsymbol{x})\Big) && && \mathcal{R} \\ s.t. \ & \sum_{j \in [n]} A_{ij} \boldsymbol{x}_j \left(1 - \boldsymbol{s}_j\right) \geq 1 && \forall i \in [m] \ \ \forall \boldsymbol{s} \in \mathcal{U} \\ & \boldsymbol{x} \in \mathcal{X} \end{aligned}$$

Let $\mathbf{f} \in \mathbb{R}_+^n$ ann let $\mathcal{Z} \subseteq \{0,1\}^n \times \{0,1\}^n$ such that $\mathcal{Z}$ is 0-1-MILP representable. The *Robust combinatorial optimization problem with fortification under locally budgeted interdiction uncertainty* *against the objective function and covering constraints* is a problem in $(\boldsymbol{x}, \boldsymbol{z})$ defined as follows:

$$\max \quad \left\{ v\Big(\mathcal{WS}(\boldsymbol{x})\Big) + \sum_{j\in[n]} (\mathbf{q}_j - \mathbf{f}_j)\boldsymbol{z}_j \right\} \qquad \mathcal{R}_f$$

$$s.t. \quad \sum_{j\in[n]} A_{ij}\Big(\boldsymbol{x}_j(1-\boldsymbol{s}) + \boldsymbol{z}_j\Big) \geq 1 \qquad \forall i \in [m] \;\; \forall \boldsymbol{s} \in \mathcal{U}$$

$$\boldsymbol{x} + \boldsymbol{z} \leq \mathbf{1}$$

$$(\boldsymbol{x}, \boldsymbol{z}) \in \mathcal{Z}$$

In the fortified problem each variable j has two versions: $\boldsymbol{x}_j$ (which can be attacked) and $\boldsymbol{z}_j$ (when it is fortified and cannot be attacked). If the variable is used it can be used in the attackable version or in the fortified version but not in both. Fortifying has a price $\mathbf{f}_j$ and in general affects the feasible region according to $\mathcal{Z}$.

Note that $\mathcal{R}$ and $\mathcal{R}_f$ may be interpreted as Combinatorial Bilinear problems ([5]), particular case of Bilevel problems, and then several algorithms based on *scenario generation* may be used ([6],[7],[5],[8]). However, our experience with $\mathcal{R}(\mathcal{R}_f)$ being solved with scenario generation algorithms was very discouraging.

The aim of the paper is to present and experiment with compact formulations for $\mathcal{R}$ and $\mathcal{R}_f$. The compact formulations, when solved with general algorithms (e.g. with CPLEX), can be useful for some data configurations and, in general, provide us with options to create specific algorithms other than those based on scenario generation.

The study of the computational complexity of $\mathcal{R}(\mathcal{R}_f)$ is beyond the scope of this paper and for that see [3]. In [3] the authors study the complexity of $\mathcal{R}$ with $R = 1$ and $\mathcal{NV} = \emptyset$ when the attackable constraints and the objective function are defined with cardinality functions. The uncertainty set in [3] is defined from a knapsack constraint.

In section 2 we present compact formulations for $\mathcal{R}$ and $\mathcal{R}_f$. Computational experience is presented in section 3. In section 4 we present a conclusions and further extensions.

2 Theorethical results and compact formulations

The remarks, lemmas and models presented for $\mathcal{R}$ can be carefully extended for $\mathcal{R}_f$ so to save space we have omitted the proofs and comments for the $\mathcal{R}_f$ case.

2.1 Feasible solution set

Remark 1 allow us to find a compact formulation for $F(\mathcal{R})$ and is a generalization for the case $R > 1$ (with covering constraints) on the result that can be seen in [3] for the case $R = 1$.

Remark 1. *Let $\boldsymbol{x} \in \{0,1\}^n$ then: $\sum_{j\in[n]} A_{ij}\boldsymbol{x}_j(1-\boldsymbol{s}_j) \geq 1$ for all $i \in [m]$ and for all $\boldsymbol{s} \in \mathcal{U}$ if and only if for all $i \in [m]$ there exists $r(i) \in [R]$ such that $\sum_{j\in J_{r(i)}} A_{ij}\mathbf{w}_j\boldsymbol{x}_j \geq \boldsymbol{\Gamma}_{r(i)} + 1$*

Proof:

- let $i \in [m]$ such that $\sum_{j\in J_r} A_{ij}\mathbf{w}_j\boldsymbol{x}_j \leq \boldsymbol{\Gamma}_r$ for all $r \in [R]$ and let $\boldsymbol{s} \in \{0,1\}^n$ with $\boldsymbol{s}_j = 1$ if and only if $A_{ij}\boldsymbol{x}_j = 1$ for all $j \in [n]$. Since $\sum_{j\in J_r} \mathbf{w}_j\boldsymbol{s}_j = \sum_{j\in J_r} A_{ij}\mathbf{w}_j\boldsymbol{x}_j \leq \boldsymbol{\Gamma}_r$ for all $r \in [R]$ we have that $\boldsymbol{s} \in \mathcal{U}$ and $\sum_{j\in[n]} A_{ij}\boldsymbol{x}_j(1-\boldsymbol{s}_j) = \sum_{j\in[n]} A_{ij}\boldsymbol{x}_j(1-A_{ij}\boldsymbol{x}_j) = 0$,

- let $\boldsymbol{s} \in \mathcal{U}$ and let $i \in [m]$ such that $\sum_{j\in[n]} A_{ij}\boldsymbol{x}_j(1-\boldsymbol{s}_j) = 0$ then $\boldsymbol{s}_j = 1$ for all $j \in [n]$ such that $A_{ij}\boldsymbol{x}_j = 1$ and we have that $\boldsymbol{\Gamma}_r \geq \sum_{j\in J_r} \mathbf{w}_j\boldsymbol{s}_j \geq \sum_{j\in J_r} A_{ij}\mathbf{w}_j\boldsymbol{x}_j$ for all $r \in [R]$ ■

Remark 1 tells us that $\boldsymbol{x}$ can be successfully defended for each i if and only if for each i there exists a subset $J_{r(i)}$ with which it can be defended. The defense with $J_{r(i)}$ for each i is performed as indicated in the result that can be seen in [3].

Remark 2. *Let $(\boldsymbol{x},\boldsymbol{z}) \in \{0,1\}^n \times \{0,1\}^n$ with $\boldsymbol{x}+\boldsymbol{z} \leq \mathbf{1}$ then: $\sum_{j\in[n]} A_{ij}\Big(\boldsymbol{x}_j(1-\boldsymbol{s}_j)+\boldsymbol{z}_j\Big) \geq 1$ for all $i \in [m]$ and for all $\boldsymbol{s} \in \mathcal{U}$ if and only if for all $i \in [m]$ there exists $r(i) \in [R]$ such that $\sum_{j\in J_{r(i)}} A_{ij}\Big(\mathbf{w}_j\boldsymbol{x}_j + (\boldsymbol{\Gamma}_{r(i)}+1)\boldsymbol{z}_j\Big) \geq \boldsymbol{\Gamma}_{r(i)} + 1$*

2.2 Optimal solution for the worst scenario problem

Remark 3 allows us to define an optimal solution for $\mathcal{WS}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \{0,1\}^n$. Let us suppose that $J_r = \{j_1^r, \cdots, j_{n_r}^r\}$ with $\mathbf{q}_{j_1^r}/\mathbf{w}_{j_1^r} \geq \cdots \geq \mathbf{q}_{j_{n_r}^r}/\mathbf{w}_{j_{n_r}^r}$ for all $r \in [R]$ and let $W >> 0$ in such a manner that if $j_l^r \in \mathcal{NV}$ then $j_k^r \in \mathcal{NV}$ for all $k \in \{l, \cdots, n_r\}$ for all $r \in [R]$. If $\boldsymbol{x} \in \{0,1\}^n$ let $\boldsymbol{p}(\boldsymbol{x})_r = \max\{p \in \{0\} \cup [n_r] : \sum_{l\in[p]} \mathbf{w}_{j_l^r}\boldsymbol{x}_{j_l^r} \leq \boldsymbol{\Gamma}_r\}$ for all $r \in [R]$.

From now on everything is written with the variables ordered as defined above. W is selected and the ties are broken in such a way that the non-vulnerable variables all appear together at the end of the sequence of each subset. Note that $p(\boldsymbol{x})_r = 0$ if in J_r all variables are non-vulnerable.

Remark 3. *Let $\boldsymbol{x} \in \{0,1\}^n$ then there is an optimal solution for $\mathcal{WS}(\boldsymbol{x})$ defined as follows:*

$$\boldsymbol{s}(\boldsymbol{x})_{j_l^r} = \begin{cases} \boldsymbol{x}_{j_l^r} & \text{if} \quad l \in [\boldsymbol{p}(\boldsymbol{x})_r] \\ 0 & \text{if} \quad \boldsymbol{p}(\boldsymbol{x})_r < l \leq n_r \end{cases} \quad \text{for all } l \in [n_r] \text{ and for all } r \in [R]$$

Proof: let $r \in [R]$ and let $\mathcal{KP}_r(\boldsymbol{x})$ be the knapsack problem in $\boldsymbol{s}$ defined as follows:

$$\begin{aligned} \max \quad & \sum_{l\in[n_r]} \mathbf{q}_{j_l^r}\boldsymbol{x}_{j_l^r}\boldsymbol{s}_{j_l^r} && \mathcal{KP}_r(\boldsymbol{x}) \\ s.t. \quad & \sum_{l\in[n_r]} \mathbf{w}_{j_l^r}\boldsymbol{s}_{j_l^r} \leq \boldsymbol{\Gamma}_r \\ & \boldsymbol{s}_{j_l^r} \in \{0,1\}\ \forall l \in [n_r] \end{aligned}$$

Note that $\min_{\boldsymbol{s}\in\mathcal{U}} \sum_{j\in[n]} \mathbf{q}_j \boldsymbol{x}_j(1-\boldsymbol{s}_j) = \sum_{j\in[n]} \mathbf{q}_j \boldsymbol{x}_j - \max_{\boldsymbol{s}\in\mathcal{U}} \sum_{j\in[n]} \mathbf{q}_j \boldsymbol{x}_j \boldsymbol{s}_j = \sum_{j\in[n]} \mathbf{q}_j \boldsymbol{x}_j - \sum_{r\in[R]} v\Big(\mathcal{KP}_r(\boldsymbol{x})\Big)$ and $\mathcal{KP}_r(\boldsymbol{x})$ is a trivial knapsack problem and there is an optimal solution defined as follows: $\boldsymbol{s}(\boldsymbol{x})_{j_l^r} = \begin{cases} \boldsymbol{x}_{j_l^r} & \text{if} \quad l \in [\boldsymbol{p}(\boldsymbol{x})_r] \\ 0 & \text{if} \quad \boldsymbol{p}(\boldsymbol{x})_r < l \leq n_r \end{cases}$ for all $l \in [n_r]$ and for all $r \in [R]$ ■

2.3 Compact formulations without fortification

Note that $\boldsymbol{s}(\boldsymbol{x})_j = \boldsymbol{x}_j \boldsymbol{s}(\boldsymbol{x})_j$ for all $j \in [n]$ and for all $\boldsymbol{x} \in \{0,1\}^n$ therefore from remarks 1 and 3 we have that $\mathcal{R}$ may be rewritten as an equivalent problem $\mathcal{R}^+$ in $(\boldsymbol{x}, \boldsymbol{y})$ defined as follows:

$$\begin{aligned}
\max \sum_{r\in[R]} \sum_{l\in[n_r]} & \mathbf{q}_{j_l^r} \Big(\boldsymbol{x}_{j_l^r} - \boldsymbol{s}(\boldsymbol{x})_{j_l^r}\Big) && \mathcal{R}^+ \\
s.t. \quad & \sum_{l\in[n_r]} A_{ij_l^r} \mathbf{w}_{j_l^r} \boldsymbol{x}_{j_l^r} \geq \Big(\boldsymbol{\Gamma}_r + 1\Big) \boldsymbol{y}_{ir} && \forall i \in [m] \ \forall r \in [R] \\
& \sum_{r\in[R]} \boldsymbol{y}_{ir} = 1 && \forall i \in [m] \\
& \boldsymbol{x} \in \mathcal{X}, \ \boldsymbol{y} \in \{0,1\}^{m\times R}
\end{aligned}$$

In order to define a *natural* compact formulation for $\mathcal{R}^+$ we need auxiliary variables and constraints to compute $\boldsymbol{s}(\boldsymbol{x})$. Let $\boldsymbol{x} \in \{0,1\}^n$ and let $\boldsymbol{v}(\boldsymbol{x}) \in \{0,1\}^n$ defined as follows:

$$\frac{\sum_{l\in[k]} \mathbf{w}_{j_l^r} \boldsymbol{x}_{j_l^r} - \boldsymbol{\Gamma}_r}{\sum_{l\in[k]} \mathbf{w}_{j_l^r} - \boldsymbol{\Gamma}_r} \leq 1 - \boldsymbol{v}(\boldsymbol{x})_{j_k^r} \leq \frac{\sum_{l\in[k]} \mathbf{w}_{j_l^r} \boldsymbol{x}_{j_l^r}}{\boldsymbol{\Gamma}_r + 1} \qquad \forall k \in [n_r] \ \forall r \in [R]$$

From definition of $\boldsymbol{v}$ and from remark 3 we have the following remark:

Remark 4. *Let $\boldsymbol{x} \in \{0,1\}^n$ then:*

1. $\boldsymbol{v}(\boldsymbol{x})_{j_k^r} = 1$ *if and only if* $\sum_{l\in[k]} \mathbf{w}_{j_l^r} \boldsymbol{x}_{j_l^r} \leq \boldsymbol{\Gamma}_r$ *for all* $\forall k \in [n_r]$ *and* $\forall r \in [R]$
2. $\boldsymbol{s}(\boldsymbol{x})_{j_k^r} = \boldsymbol{v}(\boldsymbol{x})_{j_k^r} \boldsymbol{x}_{j_k^r}$ *for all* $\forall k \in [n_r]$ *and for all* $r \in [R]$

2.3.1 Model $\mathcal{N}$

The *Natural* compact formulation for $\mathcal{R}^+$ is defined as follows:

Lemma 1. *Let $\mathcal{N}$ be a problem in $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s}, \boldsymbol{v})$ defined as follows:*

$$\max \sum_{r\in[R]} \sum_{l\in[n_r]} \mathbf{q}_{j_l^r}\Big(\boldsymbol{x}_{j_l^r} - \boldsymbol{s}_{j_l^r}\Big) \qquad \mathcal{N}$$

$$s.t. \sum_{l\in[n_r]} A_{ij_l^r}\mathbf{w}_{j_l^r}\boldsymbol{x}_{j_l^r} \geq \Big(\boldsymbol{\Gamma}_r + 1\Big)\boldsymbol{y}_{ir} \qquad \forall i \in [m]\ \forall r \in [R] \qquad \mathcal{N}.1.1$$

$$\sum_{r\in[R]} \boldsymbol{y}_{ir} = 1 \qquad \forall i \in [m] \qquad \mathcal{N}.1.2$$

$$\sum_{l\in[k]} \mathbf{w}_{j_l^r}\boldsymbol{x}_{j_l^r} \leq \boldsymbol{\Gamma}_r + \Big(1 - \boldsymbol{v}_{j_k^r}\Big)\Big(\sum_{l\in[k]} \mathbf{w}_{j_l^r} - \boldsymbol{\Gamma}_r\Big) \qquad \forall k \in [n_r]\ \forall r \in [R] \qquad \mathcal{N}.2.1$$

$$\sum_{l\in[k]} \mathbf{w}_{j_l^r}\boldsymbol{x}_{j_l^r} \geq \Big(\boldsymbol{\Gamma}_r + 1\Big)\Big(1 - \boldsymbol{v}_{j_k^r}\Big) \qquad \forall k \in [n_r]\ \forall r \in [R] \qquad \mathcal{N}.2.2$$

$$\boldsymbol{s} \geq \boldsymbol{x} + \boldsymbol{v} - \mathbf{1} \qquad \mathcal{N}.3.1$$

$$\boldsymbol{s} \leq \boldsymbol{x} \qquad \mathcal{N}.3.2$$

$$\boldsymbol{s} \leq \boldsymbol{v} \qquad \mathcal{N}.3.3$$

$$\boldsymbol{x} \in \mathcal{X},\ \boldsymbol{y} \in \{0,1\}^{m\times R},\ \boldsymbol{s} \in \{0,1\}^n,\ \boldsymbol{v} \in \{0,1\}^n$$

then $\mathcal{R}^+$ and $\mathcal{N}$ are equivalent problems

Proof: let $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s}, \boldsymbol{v}) \in F(\mathcal{N})$ then:

1. from $\mathcal{N}.1.1$ and $\mathcal{N}.1.2$ we have that $(\boldsymbol{x}, \boldsymbol{y}) \in F(\mathcal{R}^+)$,
2. from $\mathcal{N}.2.1$ and $\mathcal{N}.2.2$ we have that $\boldsymbol{v}_{j_k^r} = \boldsymbol{v}(\boldsymbol{x})_{j_k^r}$ for all $k \in [n_r]$ and for all $r \in [R]$,
3. from $\mathcal{N}.3.1$, $\mathcal{N}.3.2$, $\mathcal{N}.3.3$ and from remark 4 we have that $\boldsymbol{s}_{j_k^r} = \boldsymbol{v}_{j_k^r}\boldsymbol{x}_{j_k^r} = \boldsymbol{s}(\boldsymbol{x})_{j_k^r}$ for all $k \in [n_r]$ and for all $r \in [R]$ ■

2.3.2 Model $\mathcal{S}$

Since maximization is the optimization criterion we may delete constraints $\mathcal{N}.2.1$, $\mathcal{N}.3.2$ and $\mathcal{N}.3.3$ to obtain tha following model:

$$\max \sum_{r\in[R]} \sum_{l\in[n_r]} \mathbf{q}_{j_l^r}\Big(\boldsymbol{x}_{j_l^r} - \boldsymbol{s}_{j_l^r}\Big)$$

$$s.t. \sum_{l\in[n_r]} A_{ij_l^r}\mathbf{w}_{j_l^r}\boldsymbol{x}_{j_l^r} \geq \Big(\boldsymbol{\Gamma}_r + 1\Big)\boldsymbol{y}_{ir} \qquad \forall i\in[m]\ \forall r\in[R]$$

$$\sum_{r\in[R]} \boldsymbol{y}_{ir} = 1 \qquad \forall i\in[m]$$

$$\sum_{l\in[k]} \mathbf{w}_{j_l^r}\boldsymbol{x}_{j_l^r} \geq \Big(\boldsymbol{\Gamma}_r + 1\Big)\Big(1 - \boldsymbol{v}_{j_k^r}\Big) \qquad \forall k\in[n_r]\ \forall r\in[R]$$

$$\boldsymbol{s} \geq \boldsymbol{x} + \boldsymbol{v} - \mathbf{1}$$

$$\boldsymbol{x}\in\mathcal{X},\ \boldsymbol{y}\in\{0,1\}^{m\times R},\ \boldsymbol{s}\in\{0,1\}^n,\ \boldsymbol{v}\in\{0,1\}^n$$

which leads us after a change of variable ($\boldsymbol{u} = \boldsymbol{x} - \boldsymbol{s}$) and carefully eliminating $\boldsymbol{v}$, to the model $\mathcal{S}$ to be defined below.

Lemma 2. *Let $\mathcal{S}$ be a problem in $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u})$ defined as follows:*

$$\max \sum_{r\in[R]}\sum_{l\in[n_r]} \mathbf{q}_{j_l^r}\boldsymbol{u}_{j_l^r} \qquad \mathcal{S}$$

$$s.t. \sum_{l\in[n_r]} A_{ij_l^r}\mathbf{w}_{j_l^r}\boldsymbol{x}_{j_l^r} \geq (\boldsymbol{\Gamma}_r + 1)\boldsymbol{y}_{ir} \qquad \forall i\in[m]\ \forall r\in[R] \qquad \mathcal{S}.1.1$$

$$\sum_{r\in[R]} \boldsymbol{y}_{ir} = 1 \qquad \forall i\in[m] \qquad \mathcal{S}.1.2$$

$$\sum_{l\in[k]} \mathbf{w}_{j_l^r}\boldsymbol{x}_{j_l^r} \geq \Big(\boldsymbol{\Gamma}_r + 1\Big)\boldsymbol{u}_{j_k^r} \qquad \forall k\in[n_r]\ \ \forall r\in[R] \qquad \mathcal{S}.2.1$$

$$\boldsymbol{u} \leq \boldsymbol{x} \qquad \mathcal{S}.2.2$$

$$\boldsymbol{x}\in\mathcal{X},\ \boldsymbol{y}\in\{0,1\}^{m\times R},\ \ \boldsymbol{u}\in\{0,1\}^n$$

then $\mathcal{R}^+$ and $\mathcal{S}$ are equivalent problems

Proof:

- Let $(\boldsymbol{x}, \boldsymbol{y})$ be an optimal solution for $\mathcal{R}^+$ and let $\boldsymbol{u}(\boldsymbol{x})_{j_k^r} = \begin{cases} 0 & \text{if} \quad k\in[\boldsymbol{p}(\boldsymbol{x})_r] \\ \boldsymbol{x}_{j_k^r} & \text{if} \quad \boldsymbol{p}(\boldsymbol{x})_r < k \leq n_r \end{cases}$ for all $k\in[n_r]$ and for all $r\in[R]$ then $\Big(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{x})\Big)$ satisfies $\mathcal{S}.2.1$ and $\mathcal{S}.2.2$ and we have that $\Big(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}(\boldsymbol{x})\Big) \in F(\mathcal{S})$. From definition we have that $\boldsymbol{u}(\boldsymbol{x})_{j_l^r} = \boldsymbol{x}_{j_l^r} - \boldsymbol{s}(\boldsymbol{x})_{j_l^r}$ and then $\sum_{r\in[R]}\sum_{l\in[n_r]} \mathbf{q}_{j_l^r}\boldsymbol{u}(\boldsymbol{x})_{j_l^r} = v(\mathcal{WS}(\boldsymbol{x}))$.
- Let $\Big(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}\Big)$ be an optimal solution for $\mathcal{S}$ then $(\boldsymbol{x}, \boldsymbol{y}) \in F(\mathcal{R}^+)$. Since maximization is the optimization criterion and from $\mathcal{S}.2.1$ and $\mathcal{S}.2.2$ we have that $\Big(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}(\boldsymbol{x})\Big)$ is an optimal solution for $\mathcal{S}$ and $\sum_{r\in[R]}\sum_{l\in[n_r]} \mathbf{q}_{j_l^r}\boldsymbol{u}(\boldsymbol{x})_{j_l^r} = v\Big(\mathcal{WS}(\boldsymbol{x})\Big)$ ■

2.4 Compact formulations with fortification

$\mathcal{R}_f$ may be rewritten as an equivalent problem $\mathcal{R}_f^+$ in $(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y})$ defined as follows:

$$\begin{aligned}
\max & \sum_{r\in[R]} \sum_{l\in[n_r]} \mathbf{q}_{j_l^r}\Big(\boldsymbol{x}_{j_l^r} - \boldsymbol{s}(\boldsymbol{x})_{j_l^r}\Big) && && \mathcal{R}_f^+ \\
s.t. & \sum_{l\in[n_r]} A_{ij_l^r}\Big(\mathbf{w}_{j_l^r}\boldsymbol{x}_{j_l^r} + (\boldsymbol{\Gamma}_r + 1)\boldsymbol{z}_{j_l^r}\Big) \geq \Big(\boldsymbol{\Gamma}_r + 1\Big)\boldsymbol{y}_{ir} && \forall i \in [m]\ \forall r \in [R] \\
& \sum_{r\in[R]} \boldsymbol{y}_{ir} = 1 && \forall i \in [m] \\
& \boldsymbol{x} + \boldsymbol{z} \leq \mathbf{1} \\
& (\boldsymbol{x}, \boldsymbol{z}) \in \mathcal{Z}, \quad \boldsymbol{y} \in \{0,1\}^{m\times R}
\end{aligned}$$

2.4.1 Model $\mathcal{N}_f$

Lemma 3. *Let $\mathcal{N}_f$ be a problem in $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{s}, \boldsymbol{v})$ defined as follows:*

$$\begin{aligned}
\max & \sum_{r\in[R]} \sum_{l\in[n_r]} \Big(\mathbf{q}_{j_l^r} - \mathbf{f}_{j_l^r}\Big)\boldsymbol{z}_{j_l^r} + \sum_{r\in[R]} \sum_{l\in[n_r]} \mathbf{q}_{j_l^r}\Big(\boldsymbol{x}_{j_l^r} - \boldsymbol{s}_{j_l^r}\Big) && && \mathcal{N}_f \\
s.t. & \sum_{l\in[n_r]} A_{ij_l^r}\Big(\mathbf{w}_{j_l^r}\boldsymbol{x}_{j_l^r} + (\boldsymbol{\Gamma}_r + 1)\boldsymbol{z}_{j_l^r}\Big) \geq \Big(\boldsymbol{\Gamma}_r + 1\Big)\boldsymbol{y}_{ir} && \forall i \in [m]\ \forall r \in [R] && \mathcal{N}_f.1.1 \\
& \sum_{r\in[R]} \boldsymbol{y}_{ir} = 1 && \forall i \in [m] && \mathcal{N}_f.1.2 \\
& \sum_{l\in[k]} \mathbf{w}_{j_l^r}\boldsymbol{x}_{j_l^r} \leq \boldsymbol{\Gamma}_r + \Big(1 - \boldsymbol{v}_{j_k^r}\Big)\Big(\sum_{l\in[k]} \mathbf{w}_{j_k^r} - \boldsymbol{\Gamma}_r\Big) && \forall k \in [n_r]\ \forall r \in [R] && \mathcal{N}_f.2.1 \\
& \sum_{l\in[k]} \mathbf{w}_{j_l^r}\boldsymbol{x}_{j_l^r} \geq \Big(\boldsymbol{\Gamma}_r + 1\Big)\Big(1 - \boldsymbol{v}_{j_k^r}\Big) && \forall k \in [n_r]\ \forall r \in [R] && \mathcal{N}_f.2.2 \\
& \boldsymbol{s} \geq \boldsymbol{x} + \boldsymbol{v} - \mathbf{1} && && \mathcal{N}_f.3.1 \\
& \boldsymbol{s} \leq \boldsymbol{x} && && \mathcal{N}_f.3.2 \\
& \boldsymbol{s} \leq \boldsymbol{v} && && \mathcal{N}_f.3.3 \\
& \boldsymbol{x} + \boldsymbol{z} \leq \mathbf{1} && && \mathcal{N}_f.4 \\
& (\boldsymbol{x}, \boldsymbol{z}) \in \mathcal{Z},\ \boldsymbol{y} \in \{0,1\}^{m\times R},\ \boldsymbol{s} \in \{0,1\}^n,\ \boldsymbol{v} \in \{0,1\}^n
\end{aligned}$$

then $\mathcal{R}_f^+$ and $\mathcal{N}_f$ are equivalent problems

2.4.2 Model $\mathcal{S}_f$

Lemma 4. *Let $\mathcal{S}_f$ be a problem in $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{u})$ defined as follows:*

$$\max \sum_{r\in[R]}\sum_{l\in[n_r]} \mathbf{q}_{j_l^r} \boldsymbol{u}_{j_l^r} + \sum_{r\in[R]}\sum_{l\in[n_r]} \left(\mathbf{q}_{j_l^r} - \mathbf{f}_{j_l^r}\right)\boldsymbol{z}_{j_l^r} \qquad \mathcal{S}_f$$

$$s.t. \sum_{l\in[n_r]} A_{ij_l^r}\left(\mathbf{w}_{j_l^r}\boldsymbol{x}_{j_l^r} + (\boldsymbol{\Gamma}_r + 1)\boldsymbol{z}_{j_l^r}\right) \geq (\boldsymbol{\Gamma}_r + 1)\boldsymbol{y}_{ir} \qquad \forall i \in [m]\ \forall r \in [R] \qquad \mathcal{S}_f.1.1$$

$$\sum_{r\in[R]} \boldsymbol{y}_{ir} = 1 \qquad \forall i \in [m] \qquad \mathcal{S}_f.1.2$$

$$\sum_{l\in[k]} \mathbf{w}_{j_l^r}\boldsymbol{x}_{j_l^r} \geq \left(\boldsymbol{\Gamma}_r + 1\right)\boldsymbol{u}_{j_k^r} \qquad \forall k \in [n_r]\ \ \forall r \in [R] \qquad \mathcal{S}_f.2.1$$

$$\boldsymbol{u} \leq \boldsymbol{x} \qquad \mathcal{S}.2.2$$

$$\boldsymbol{x} + \boldsymbol{z} \leq \mathbf{1} \qquad \mathcal{S}_f.3$$

$$(\boldsymbol{x}, \boldsymbol{z}) \in \mathcal{Z},\ \boldsymbol{y} \in \{0,1\}^{m\times R},\ \ \boldsymbol{u} \in \{0,1\}^n$$

then $\mathcal{R}_f^+$ and $\mathcal{S}_f$ are equivalent problems

2.5 Linear relaxations

Lemma 5. *Let $\overline{\mathcal{N}}$ and $\overline{\mathcal{S}}$ be the linear relaxations for $\mathcal{N}$ and $\mathcal{S}$ respectively then $v(\overline{\mathcal{N}}) = v(\overline{\mathcal{S}})$*

Proof:

1. Let $(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}, \overline{\boldsymbol{s}}, \overline{\boldsymbol{v}})$ be an optimal solution for $\overline{\mathcal{N}}$ and let $\overline{\boldsymbol{u}} = \overline{\boldsymbol{x}} - \overline{\boldsymbol{s}}$ then:
   (a) from trivial algebraic manipulations we have that $(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}, \overline{\boldsymbol{u}}) \in F(\overline{S})$,
   (b) $v(\overline{\mathcal{N}}) = \sum_{r\in[R]}\sum_{l\in[n_r]} \mathbf{q}_{j_l^r}\left(\overline{\boldsymbol{x}}_{j_l^r} - \overline{\boldsymbol{s}}_{j_l^r}\right) = \sum_{r\in[R]}\sum_{l\in[n_r]} \mathbf{q}_{j_l^r}\overline{\boldsymbol{u}}_{j_l^r} \leq v(\overline{\mathcal{S}})$.

2. Let $(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}, \overline{\boldsymbol{u}})$ be an optimal solution for $\overline{\mathcal{S}}$ and let:
   - $\overline{\boldsymbol{s}} = \overline{\boldsymbol{x}} - \overline{\boldsymbol{u}}$
   - $\overline{\boldsymbol{v}}_{j_k^r} = 1 - \min\left\{1, \sum_{l\in[k]} \mathbf{w}_{j_l^r}\overline{\boldsymbol{x}}_{j_l^r} / \left(\Gamma_r + 1\right)\right\}$ for all $k \in [n_r]$ and for all $r \in [R]$ then:

   (a) from trivial algebraic manipulations we have that $(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}, \overline{\boldsymbol{s}}, \overline{\boldsymbol{v}}) \in F(\overline{\mathcal{N}})$,
   (b) $v(\overline{\mathcal{S}}) = \sum_{r\in[R]}\sum_{l\in[n_r]} \mathbf{q}_{j_l^r}\overline{\boldsymbol{u}}_{j_l^r} = \sum_{r\in[R]}\sum_{l\in[n_r]} \mathbf{q}_{j_l^r}\left(\overline{\boldsymbol{x}}_{j_l^r} - \overline{\boldsymbol{s}}_{j_l^r}\right) \leq v(\overline{\mathcal{N}})$ ■

Lemma 6. *Let $\overline{\mathcal{N}_f}$ and $\overline{\mathcal{S}_f}$ be the linear relaxations for $\mathcal{N}_f$ and $\mathcal{S}_f$ respectively then $v(\overline{\mathcal{N}_f}) = v(\overline{\mathcal{S}_f})$*

2.6 Minor remarks

- Let $(\boldsymbol{x},\boldsymbol{y}) \in F(\mathcal{R}^+)$. Note that if $(\boldsymbol{x},\boldsymbol{y},\boldsymbol{s},\boldsymbol{v}) \in F(\mathcal{N})$ then $\boldsymbol{s} = \boldsymbol{s}(\boldsymbol{x})$. There is no an analogous result with $\mathcal{S}$ because of $(\boldsymbol{x},\boldsymbol{y},\boldsymbol{0}) \in F(\mathcal{S})$ for all $(\boldsymbol{x},\boldsymbol{y}) \in F(\mathcal{R}^+)$. For an optimal solution for $\mathcal{S}$ we have that $\boldsymbol{u} = \boldsymbol{x} - \boldsymbol{s}(\boldsymbol{x})$.

- Let $(\boldsymbol{x},\boldsymbol{z},\boldsymbol{y}) \in F(\mathcal{R}_f^+)$. Note that if $(\boldsymbol{x},\boldsymbol{z},\boldsymbol{y},\boldsymbol{s},\boldsymbol{v}) \in F(\mathcal{N}_f)$ then $\boldsymbol{s} = \boldsymbol{x}(\boldsymbol{x})$. There is no an analogous result with $\mathcal{S}_f$ because of $(\boldsymbol{x},\boldsymbol{z},\boldsymbol{y},\boldsymbol{0}) \in F(\mathcal{S}_f)$ for all $(\boldsymbol{x},\boldsymbol{z},\boldsymbol{y}) \in F(\mathcal{R}_f^+)$. For an optimal solution for $\mathcal{S}_f$ we have that $\boldsymbol{u} = \boldsymbol{x} - \boldsymbol{s}(\boldsymbol{x})$.

- Since maximization is the optimization criterium we may delete $\mathcal{N}$.3.2 and $\mathcal{N}$.3.3 ($\mathcal{N}_f$.3.2 and $\mathcal{N}_f$.3.3) in order to define an additional compact formultation, $\mathcal{N}^-(\mathcal{N}_f^-)$, to be used in our experiments, with the same linear relaxation optimal values ($v(\overline{\mathcal{N}}) = v(\overline{\mathcal{N}^-}) = v(\overline{\mathcal{S}})$ and $v(\overline{\mathcal{N}_f}) = v(\overline{\mathcal{N}_f^-}) = v(\overline{\mathcal{S}_f})$).

3 Computational experience

The experiments have been performed on a personal computer as follows: Intel(R)Core(TM) i7-9750H CPU, @ 2.60 GHz Lenovo ThinkPad X1 Extreme Gen 2, 32.00 GB Ram and Windows 10 Pro Operating System. All the instances have been processed through ILOG-Cplex 12.10 from a DOcplex Python code. All the parameters of ILOG-Cplex 12.10 are in their default values, including the required *gap* (0.01%), except for the *integrality tolerance parameter* for which we use 10^{-12}.

In section 3.1 we explain how the problems were generated and in section 3.2 we present the experiments performed and the results.

3.1 Problems and data generation

We proceed as follows:

1. let $R \geq 1$, let $L \geq 1$, let $n = RL$, let $n_r = L$ and let $j_l^r = (r-1)L + l$ for all $l \in [L]$, for all $r \in [R]$,

2. we experiment with two types of problems according to A as follows:

   (a) i. let $m = n$,

   ii. let $\Delta = 1/R$, L points are generated for each $r \in [R]$ with randomly uniform angle at $[0, 2\pi]$ and randomly uniform radius at $[(r-1)\Delta, r\Delta]$,

   iii. let $\mathbf{d}_{ij}$ be the Euclidean distance between points i and j and let $dmax = \max_{(i,j)\in[n]\times[n]} \mathbf{d}_{ij}$,

   iv. let $\sigma_A \in (0,1)$ and let $A \in \{0,1\}^{n\times n}$ defined as follows: $A_{ij} = 1$ if and only if $\mathbf{d}_{ij} < \sigma_A dmax$

   (b) i. let $m = R$,

   ii. let $A \in \{0,1\}^{m\times n}$ defined as follows: $A_{rj} = 1$ if and only if $j \in J_r$ for all $r \in [m]$, for all $j \in [n]$,

3. let $\mathbf{c} \in \mathbb{R}^n_+$ be the *cost vector*, let $B > 0$ the *budget* and let $\mathcal{X} = \Big\{ \boldsymbol{x} \in \{0,1\}^n : \sum_{j \in [n]} \mathbf{c}_j \boldsymbol{x}_j \leq B \Big\}$,

4. we generate *weakly correlated instances* as follows (see [9]): let $\mathbf{q}_j$ randomly uniform in $[1000]$ for all $j \in [n]$, then we choose $\widehat{\mathbf{c}}_j$ from $\Big\{ \mathbf{q}_j - 100, \cdots, \mathbf{q}_j + 100 \Big\}$ randomly uniform and finally $\mathbf{c}_j = \max\Big\{ 1, \widehat{\mathbf{c}}_j \Big\}$ for all $j \in [n]$,

5. let $\gamma \in \mathbb{N}$, we use $\mathbf{\Gamma}_r = \gamma$ for all $r \in [R]$,

6. let $p_{nv} \in [0,1)$, in order to generate problems with $\mathcal{NV} \neq \emptyset$ $\mathbf{w}_j$ is redefined with $\mathbf{w}_j = W = 1001$ with probability p_{nv} for all $j \in [n]$,

7. let $\sigma_B \geq 1$ and let $\mathcal{B}$ be a problem in $(\boldsymbol{x}, \boldsymbol{y})$ defined as follows:

$$
\begin{aligned}
\min \ & \sum_{j \in [n]} \mathbf{c}_j \boldsymbol{x}_j && && \mathcal{B} \\
s.t. \ & \sum_{j \in J_r} A_{ij} \mathbf{w}_j \boldsymbol{x}_j \geq (\mathbf{\Gamma}_r + 1) \boldsymbol{y}_{ir} && \forall i \in [m] \ \forall r \in [R] \\
& \sum_{r \in [R]} \boldsymbol{y}_{ir} = 1 && \forall i \in [m] \\
& \boldsymbol{x} \in \{0,1\}^n, \ \boldsymbol{y} \in \{0,1\}^{m \times R}
\end{aligned}
$$

we solve $\mathcal{B}$ and then:

   (a) if $F(\mathcal{B}) \neq \emptyset$ then let $B = \sigma_B v(\mathcal{B})$,

   (b) otherwise we have that $F(\mathcal{R}^+) = \emptyset$ and the data are discarded,

8. in order to generate problems with fortification we proceed as follows: we use $\mathbf{f} = \mathbf{0}$, let $\sigma_f > 1$, let $\mathbf{F} = \sigma_f \mathbf{c}$ be the *fortification cost vector* and let $\mathcal{Z} = \Big\{ (\boldsymbol{x}, \boldsymbol{z}) \in \{0,1\}^n \times \{0,1\}^n : \sum_{j \in [n]} \Big(\mathbf{c}_j \boldsymbol{x}_j + \mathbf{F}_j \boldsymbol{z}_j \Big) \leq B \Big\}$

9. after the data are generated they are reordered in such a way that $\mathbf{q}_{j_1^r} / \mathbf{w}_{j_1^r} \geq \cdots \geq \mathbf{q}_{j_{n_r}^r} / \mathbf{w}_{j_{n_r}^r}$ for all $r \in [R]$

If we use **2.a** the $\mathcal{P}$ problem generated is a variant of a *combined knapsack-set covering problem* (see [1]) and we call it KC problem. In that case A is generated based on distances as is often done when generating instances for the *set covering* problem. If we use **2.b** the $\mathcal{P}$ problem generated is a variant of a *multiple choice knapsack problem* (see [2]) and we call it Km problem. Fortified models are KC_f and Km_f.

For Km problems generated we have that $\boldsymbol{y}_{rr} = 1$ is the only choice and we can delete the $\boldsymbol{y}$ variables and the constraints $*.1.1, *.1.2$ may be reduced to m constraints for all models.

Table 1: Sizes of models with: $A \in \{0,1\}^{m\times n}$, $\mathcal{X} = \{\boldsymbol{x} \in \{0,1\}^n : \sum_{j\in[n]} \mathbf{c}_j \boldsymbol{x}_j \le B\}$ and $\mathcal{Z} = \{(\boldsymbol{x},\boldsymbol{z}) \in \{0,1\}^n \times \{0,1\}^n : \sum_{j\in[n]}(\mathbf{c}_j\boldsymbol{x}_j + \mathbf{F}_j\boldsymbol{z}_j) \le B\}$. For all experiments $|J_r| = L$ for all $r \in [R]$ and $n = RL$. For KC problems $m = n$ and for Km problems $m = R$. For Km problems we delete $\boldsymbol{y}$ and $*.1.1, *.1.2$ reduces to m constraints for all model *

Model	# Const.	# 0-1	Model	# Const.	# 0-1	Model	# Const.	# 0-1
$\mathcal{P}$	m+1	n	KC	$RL+1$	RL	Km	$R+1$	RL
$\mathcal{S}$	$m(R+1)+\mathbf{2}n+1$	$mR+\mathbf{2}n$	$\mathcal{U}$	$L(R^2+\mathbf{3}R)+1$	$L(R^2+\mathbf{2}R)$	$\mathcal{U}$	$R(\mathbf{2}L+1)+1$	$\mathbf{2}RL$
$\mathcal{N}^-$	$m(R+1)+\mathbf{3}n+1$	$mR+\mathbf{3}n$	$\mathcal{N}^-$	$L(R^2+\mathbf{4}R)+1$	$L(R^2+\mathbf{3}R)$	$\mathcal{N}^-$	$R(\mathbf{3}L+1)+1$	$\mathbf{3}RL$
$\mathcal{N}$	$m(R+1)+\mathbf{5}n+1$	$mR+\mathbf{3}n$	$\mathcal{N}$	$L(R^2+\mathbf{6}R)+1$	$L(R^2+\mathbf{3}R)$	$\mathcal{N}$	$R(\mathbf{5}L+1)+1$	$\mathbf{3}RL$
$\mathcal{S}_f$	$m(R+1)+\mathbf{3}n+1$	$mR+\mathbf{3}n$	$\mathcal{U}_f$	$L(R^2+\mathbf{4}R)+1$	$L(R^2+\mathbf{3}R)$	$\mathcal{U}_f$	$R(\mathbf{3}L+1)+1$	$\mathbf{3}RL$
$\mathcal{N}_f^-$	$m(R+1)+\mathbf{4}n+1$	$mR+\mathbf{4}n$	$\mathcal{N}_f^-$	$L(R^2+\mathbf{5}R)+1$	$L(R^2+\mathbf{4}R)$	$\mathcal{N}_f^-$	$R(\mathbf{4}L+1)+1$	$\mathbf{4}RL$
$\mathcal{N}_f$	$m(R+1)+\mathbf{6}n+1$	$mR+\mathbf{4}n$	$\mathcal{N}_f$	$L(R^2+\mathbf{7}R)+1$	$L(R^2+\mathbf{4}R)$	$\mathcal{N}_f$	$R(\mathbf{6}L+1)+1$	$\mathbf{4}RL$

.

Note that: (i) if $\sigma_f = 1$ then $\mathcal{P}, \mathcal{N}_f$ and $\mathcal{S}_f$ are equivalent problems, (ii) we use $W = 1001$ in order to be sure that for all $j \in \mathcal{NV}$ and for all $k \in [n] - \mathcal{NV}$ we have that $\mathbf{q}_j/\mathbf{w}_j = \mathbf{q}_j/W < 1 \le \mathbf{q}_k = \mathbf{q}_k/\mathbf{w}_k$, (iii) $v(\mathcal{B})$ is an upper bound for the minimal budget for the fortification case and we use it in order to save experimental time (in such cases fortification is used to improve the quality of solutions to problems that are certainly feasible).

We use 6 generation parameters: $(p_{nv}, \gamma, \sigma_B, \sigma_A, R, L)$ to genenerate KC problems and 5 generation parameters $(p_{nv}, \gamma, \sigma_B, R, L)$ to generate Km problems. Also, we need an additional parameter σ_f to generate problems with fortification.

In table 1 we present the number of constraints (# Const.) and 0-1-variables (# 0-1) for the models $\mathcal{P}$, $\mathcal{N}$, $\mathcal{N}^-$, $\mathcal{S}$, $\mathcal{N}_f$ $\mathcal{N}_f^-$ and $\mathcal{S}_f$. Also we present # Const. and # 0-1 for the particular cases considered in the computational experience with KC and mK the nominal problems according the generation procedure.

3.2 Experiments and results

Note that with 5(6)(7) generation parameters the number of possible configurations considering several levels per parameter makes a very detailed experimentation very difficult. For problems without(with) fortification 12(4) basic configurations were arbitrarily set.

In table 2 we show the configurations used leaving one parameter free which varies as indicated. We present: the configuration number (fig), the model considered, the value of generation parameters $(pnv, \gamma, \sigma_B, \sigma_A, R, L, \sigma_f)$ ($*$ indicates that parameter is free and $-$ indicates that the parameter is not active in the generation process), the free parameter (*) and its values (with at most 8 values). From now on, $par(i)$ denotes the cardinality of the set of values that the free parameter can take in the configuration i.

For example for fig-2:

- the model used is KC,
- the parameters to generate the problems are $(pnv, \gamma, \sigma_B, \sigma_A, R, L, \sigma_f) = (0.00, 2, 1.25, *, 10, 50, -)$,

- the free parameter is σ_A,
- the parameter σ_f is not used,
- $\sigma_A \in \{0.25, 0.30, 0.35, 0.40.0.45, 0.50\}$ with $par(2) = 6$.

For each configuration and free parameter, 10 random problems are generated. We present results about 870(270) problems without(with) fortification. The results associated with each configuration are presented in Table 3 and in the figures (one figure for each configuration).

In figure-i ($i \in [16]$) we present the average (at) and maximun (mt) run time in seconds (including the time to generate the model to be solved with CPLEX) for each batch of 10 problems generated using fig_i, for each value of the free parameter and for each compact formulation. Semilogarithmic scale is used in all cases.

Black boxes in the figures indicate that the time limit was reached without a solution with the *gap* required for at least one problem in the batch and in that cases we indicate: the parameter value (p), the number of fails with formulation $\mathcal{N}^-$ ($n_{\mathcal{N}^-}$) and the number of fails with formulation $\mathcal{S}$ ($n_{\mathcal{S}}$) with the notation: $(p), \mathcal{N}^- : n_{\mathcal{N}^-}, \mathcal{S} : n_{\mathcal{S}}$. For fortification problems we use the same notation: $(p), \mathcal{N}_f^- : n_{\mathcal{N}_f^-}, \mathcal{S}_f : n_{\mathcal{S}_f}$.

For example in figure 14 we present the results for fig-14 according table 2 for the model Km_f and the problems solved using $\mathcal{S}_f$ and $\mathcal{N}_f^-$. With the blue(red) lines we present at and mt obtained with $\mathcal{S}_f(\mathcal{N}_f^-)$. When the free parameter σ_f takes the value 3.00 the required *gap* was not reached before the time limit for 3 problems using $\mathcal{S}_f$ and 4 problems using $\mathcal{N}_f^-$.

In Table 3 we present the data with which the figures were constructed: $at*_{ij}$ (average time) and $mt*_{ij}$ (worst time) with $* \in \{\mathcal{S}, \mathcal{N}, \mathcal{S}_f, \mathcal{N}_f\}$, $i \in [16]$ and $j \in [par(i)]$. For example $at\mathcal{S}_{3,4} = 430$ means that the average time using $\mathcal{S}$ for problems generated with the configuration 3 and $\sigma_B = 1.30$ was 430 seconds.

In Table 4 we present the relative percentage differences in the average (ra_{ij}) and maximum (rm_{ij}) computation times (for all $i \in [12]$, for all $j \in [par(i)]$) defined as follows:

- $ra_{ij} = \begin{cases} 100\frac{at\mathcal{S}_{ij} - at\mathcal{N}_{ij}^-}{min\{at\mathcal{S}_{ij}, at\mathcal{N}_{ij}^-\}} & \text{if} \quad max\{mt\mathcal{S}_{ij}, mt\mathcal{N}_{ij}^-\} \geq 120,\ 100\frac{\left|at\mathcal{S}_{ij} - at\mathcal{N}_{ij}^-\right|}{min\{at\mathcal{S}_{ij}, at\mathcal{N}_{ij}^-\}} \geq 10 \\ 0 & \text{otherwise} \end{cases}$
- $rm_{ij} = \begin{cases} 100\frac{mt\mathcal{S}_{ij} - mt\mathcal{N}_{ij}^-}{min\{mt\mathcal{S}_{ij}, mt\mathcal{N}_{ij}^-\}} & \text{if} \quad max\{mt\mathcal{S}_{ij}, mt\mathcal{N}_{ij}^-\} \geq 120,\ 100\frac{\left|mt\mathcal{S}_{ij} - mt\mathcal{N}_{ij}^-\right|}{min\{mt\mathcal{S}_{ij}, mt\mathcal{N}_{ij}^-\}} \geq 10 \\ 0 & \text{otherwise} \end{cases}$

Note that for $i \in \{13, 14, 15, 16\}$ $\mathcal{S}_f$ and $\mathcal{N}_f^-$ are used instead of $\mathcal{S}$ and $\mathcal{N}^-$ to define ra_{ij} and ma_{ij}.

Note that we only used significant differences (greater than 10%) and only for free parameters where the worst time for one of the formulations exceeded two minutes. Negative values (in bold) indicate that the $\mathcal{S}(\mathcal{S}_f)$ formulation performed better than the $\mathcal{N}(\mathcal{N}_f)$ formulation.

For example $ra_{16,3} = \mathbf{-56}$ presented in Table 4 means that for configuration 16 and parameter with index 3 ($\sigma_f = 1.5$) the $\mathcal{S}_f$ formulation had an average calculation time 56% better than $\mathcal{N}_f^-$.

With the information presented in the tables and figures, we can affirm that:

.

1. For the average times $\mathcal{S}(\mathcal{S}_f)$ was clearly superior for problems in configurations 2,5 and 9(14 and 16) and $\mathcal{N}^-$ was clearly superior for problems in configurations 1,4,8 and 12. For the remaining configurations (3,6,7,10,11,13 and 15) there is no clear winner.

2. For the maximun times $\mathcal{S}(\mathcal{S}_f)$ was clearly superior for problems in configurations 2(16) and $\mathcal{N}^-$ was clearly superior for data in configurations 4,6 and 12. For the remaining configurations there is no clear winner.

3. For any of the configurations it is clear that for the index of the free parameter, varying by an appreciable subset of $[par(*)]$, it was possible to solve the problems reaching the required *gap* (0.01%) with times ranging from a few seconds to a maximum of 2 hrs.

4. For 10 of the 16 configurations, the free parameter took values, generally at extreme values, for which there were problems where the required *gap* could not be reached before the 7200 second time limit.

5. For 29(31) of the 870 problems solved without fortification the required *gap* could not be reached using $\mathcal{S}(\mathcal{N}^-)$. For 6(7) of the 270 problems solved with fortification the required *gap* could not be reached using $\mathcal{S}_f(\mathcal{N}_f^-)$. This should not be taken as a sign of remarkable success, since it is obvious that by properly manipulating the free parameters one can generate many problems where one is going to fail due to the imposed limit time.

6. It is striking how extremely sensitive the performance of the formulations is with respect to the number of non-vulnerable variables: a few non-vulnerable variables were enough to move from extremely difficult problems to very easy ones to solve.

7. The minimum value of B necessary and sufficient to guarantee the viability of $\mathcal{R}$ turned out to be surprisingly low for an appreciable number of problems, which can bias, for better or worse, the evaluation of the performance of the formulations, so it is necessary to experiment with configurations that lead to higher levels of B.

4 Conclusions and further extensions

We present compact formulations and experimental results for the Robust combinatorial optimization problem under locally budgeted interdiction uncertainty against the objective function and

Table 2: Configurations used for problems generation

fig	Model	pnv	γ	σ_B	σ_A	R	L	σ_f	*	1	2	3	4	5	6	7	8
1	KC	0.00	1	1.25	0.25	*	R	-	R	15	20	25	30	35	40	45	50
2		0.00	2	1.25	*	10	50	-	σ_A	0.25	0.30	0.35	0.40	0.45	0.50	-	-
3		0.00	1	*	0.25	30	30	-	σ_B	1.00	1.10	1.20	1.30	1.40	1.50		
4		*	2	1.15	0.25	30	30	-	pnv	0.025	0.05	0.075	0.10	0.15	0.20	0.25	0.30
5		0.05	*	1.25	0.25	5	80	-	γ	1	2	3	4	5	6	7	8
6		0.10	5	1.25	0.25	30	*	-	L	30	40	50	60	70	80	90	100
7	Km	0.00	2	*	-	50	50	-	σ_B	1.00	1.05	1.10	1.15	1.20	-	-	-
8		0.00	2	1.10	-	*	R	-	R	40	50	60	70	80	90	100	-
9		*	2	1.15	-	50	50	-	pnv	0.00	0.025	0.05	0.10	0.15	0.20	0.25	0.30
10		0.00	*	1.05	-	50	50	-	γ	1	2	3	4	5	-	-	-
11		0.025	*	1.10	-	100	100	-	γ	1	2	3	4	5	6	7	8
12		*	5	1.15	-	100	100	-	pnv	0.00	0.025	0.05	0.10	0.15	0.20	0.25	0.30
13	Km_f	0.00	2	1.10	-	*	R	10	R	10	15	20	25	30	35	40	-
14		0.00	2	1.10	-	30	30	*	σ_f	1.00	1.50	2.00	2.50	3.00	-	-	-
15	KC_f	0.00	1	1.25	0.25	*	R	10	R	10	15	20	25	30	35	40	45
16		0.00	2	1.25	0.25	20	20	*	σ_f	1.00	1.25	1.50	1.75	2.00	2.25	2.50	-

covering constraints with and without nonvulnerble variables and with and without fortification. As far as we know these are the first theorethical and experimental results presented for the robust problem considered.

With the computational experience using the compact formulations we show evidence that: (i) moderate dimensional problems can be solved, (ii) by altering the data even slightly with reasonable criteria we can move from easily solvable problems to problems requiring considerable computational work, (iii) none of the formulations dominates the other in terms of computational time and this can be attributed to multiple causes (except how tight the linear relaxation is) ([10],[11]).

The path we have set out to try to find algorithms that improve the performance of the presented compact formulations includes considering the following options:

- define exact or approximate algorithms based on *Local Branching* ([12]) in which compact formulations restricted to selected neighborhoods are used,

- define *dynamic programming algorithms* ([13]) for cases where this is possible, for example: problems without coverage constraints where only the objective function is attacked and Km problems. In such cases, compact formulations will allow us the computation of the values needed to fill the network of states and decisions typical of dynamic programming.

References

[1] S. A. Mochocki, G. B. Lamont, R. C. Leishman, K. J. Kauffman, Multiobjective database queries in combined knapsack and set covering problem domains, Journal of Big Data 8:46 (2021). doi:https://doi.org/10.1186/s40537-021-00433-x.

Table 3: average times (at) and maximun times (mt) according model (*), figure (i) and parameter index (j)

*	$i \setminus j$	$at*_{ij}$ 1	2	3	4	5	6	7	8	$mt*_{ij}$ 1	2	3	4	5	6	7	8
$\mathcal{S}$	1	17	134	270	415	535	580	819	1081	26	358	807	1401	2244	1761	1626	2017
$\mathcal{N}^-$		21	138	192	361	359	403	614	885	35	433	480	1318	1099	1159	1391	2933
$\mathcal{S}$	2	3462	1036	254	91	30	13	-	-	7200	3274	1228	235	67	346	-	-
$\mathcal{N}^-$		4586	1589	594	131	34	15	-	-	7200	7200	4142	369	97	39	-	-
$\mathcal{S}$	3	86	139	343	430	466	1158	-	-	194	261	1100	2090	1838	7070	-	-
$\mathcal{N}^-$		87	157	340	380	416	1275	-	-	203	267	1208	1981	1800	7200	-	-
$\mathcal{S}$	4	2485	568	187	64	45	26	26	19	6249	2042	621	142	105	46	47	24
$\mathcal{N}^-$		2292	389	91	47	33	23	26	20	6635	1414	195	108	47	31	31	27
$\mathcal{S}$	4	14	188	506	617	447	313	114	216	30	993	2800	1465	2531	1077	319	1035
$\mathcal{N}^-$		17	189	810	689	593	382	146	237	50	871	3626	1223	3082	973	376	427
$\mathcal{S}$	6	20	54	142	171	459	361	1145	599	24	168	236	264	1568	839	7200	2310
$\mathcal{N}^-$		33	71	108	153	278	302	795	391	42	78	142	181	1073	669	4763	642
$\mathcal{S}$	7	1	26	127	1023	5205	-	-	-	1	39	238	7200	7200	-	-	-
$\mathcal{N}^-$		1	16	88	905	5989	-	-	-	2	31	163	6029	7200	-	-	-
$\mathcal{S}$	8	87	232	155	348	222	288	257	-	163	1472	363	1779	447	851	492	-
$\mathcal{N}^-$		72	107	139	215	221	205	204	-	142	336	420	702	481	622	573	-
$\mathcal{S}$	9	1206	53	22	10	7	4	4	3	7200	83	45	12	16	7	5	4
$\mathcal{N}^-$		1394	59	22	7	7	5	4	4	7200	177	39	10	13	6	6	6
$\mathcal{S}$	10	3	36	374	2978	5390	-	-	-	5	98	712	7200	7200	-	-	-
$\mathcal{N}^-$		3	18	211	2302	6733	-	-	-	4	31	414	7200	7200	-	-	-
$\mathcal{S}$	11	15	117	380	236	347	327	339	353	19	321	687	530	693	910	661	606
$\mathcal{N}^-$		16	61	155	294	249	184	177	203	24	121	425	733	660	273	360	360
$\mathcal{S}$	12	7200	1773	226	61	28	24	21	15	7200	6346	526	154	36	38	34	24
$\mathcal{N}^-$		5924	843	121	38	21	19	16	15	7200	2047	192	51	28	31	18	16
$\mathcal{S}_f$	13	3	11	32	94	297	1658	2964	-	4	24	63	436	1698	7200	7200	-
$\mathcal{N}_f^-$		3	15	48	83	411	1140	2251	-	4	32	133	228	2841	3927	7200	-
$\mathcal{S}_f$	14	1	40	857	1789	3274	-	-	-	3	77	5093	6946	7200	-	-	-
$\mathcal{N}_f^-$		1	23	1299	2719	4140	-	-	-	2	87	6100	7200	7200	-	-	-
$\mathcal{S}_f$	15	7	33	142	224	530	440	488	698	12	57	319	648	1681	1142	866	921
$\mathcal{N}_f^-$		8	40	178	230	533	426	434	598	13	74	446	655	1252	1120	741	736
$\mathcal{S}_f$	16	21	46	126	461	994	1093	2195	-	27	64	224	1139	2093	2085	6189	-
$\mathcal{N}_f^-$		19	37	197	909	1919	3080	3474	-	20	45	387	1976	3803	5528	7200	-

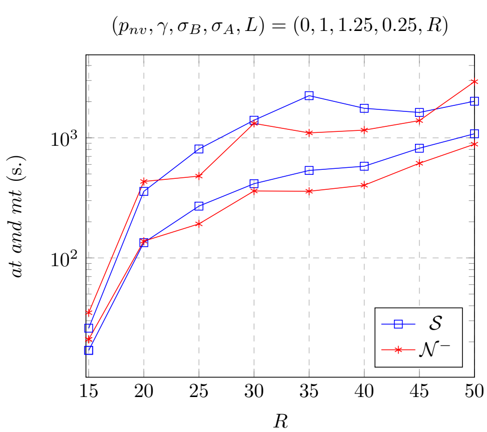

Figure 1: KC

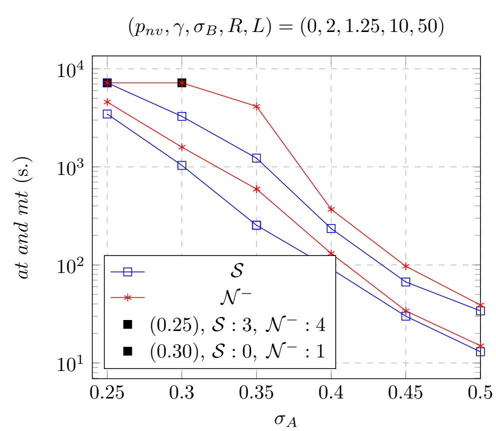

Figure 2: KC

Table 4: Relative percentage differences for average and maximun times. Negative values (in bold) indicate that the $\mathcal{S}$ ($\mathcal{S}_f$) formulation performed better that $\mathcal{N}^-$ ($\mathcal{N}_f^-$). Value 0 indicate that the difference is not significative or the maximun time for the batch was not greater than 2 minutes.

	ra								*rm*							
$i \setminus j$	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	40	14	49	43	33	22	0	**-20**	68	0	104	51	16	**-45**
2	**-32**	**-53**	**-133**	**-43**	0	0	-	-	0	**-119**	**-237**	**-57**	0	0	-	-
3	0	**-12**	0	13	12	**-10**	-	-	0	0	0	0	0	0	-	-
4	0	46	105	36	0	0	0	0	0	44	218	31	0	0	0	0
5	0	0	**-60**	**-11**	**-32**	**-22**	**-28**	0	0	14	**-29**	19	**-21**	10	**-17**	142
6	0	**-31**	31	11	65	19	44	53	0	115	66	45	46	25	51	259
7	0	0	44	13	**-15**	-	-	-	0	0	-	-	0	-	-	-
8	20	116	11	61	0	40	25	-	14	338	**-15**	153	0	36	**-16**	-
9	**-15**	**-11**	0	0	0	0	0	0	0	**-113**	0	0	0	0	0	0
10	0	0	77	29	**-24**	-	-	-	0	0	71	0	0	-	-	-
11	0	91	145	**-24**	39	77	91	73	0	165	61	**-38**	0	233	83	68
12	21	110	86	60	0	0	0	0	210	173	201	0	0	0	0	0
13	0	0	**-50**	13	**-38**	45	31	-	0	0	**-111**	91	**-67**	83	0	-
14	0	0	**-51**	**-51**	**-26**	-	-	-	0	0	**-19**	0	0	-	-	-
15	0	0	**-25**	0	0	0	12	16	0	0	**-39**	0	34	0	16	25
16	0	0	**-56**	**-97**	**-93**	**-181**	**-58**	-	0	0	**-72**	**-73**	**-81**	**-165**	**-16**	-

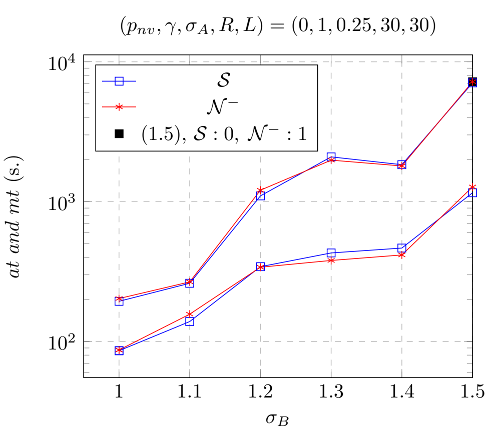

Figure 3: *KC*

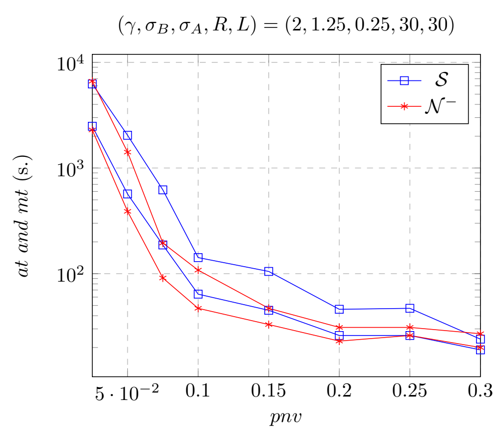

Figure 4: *KC*

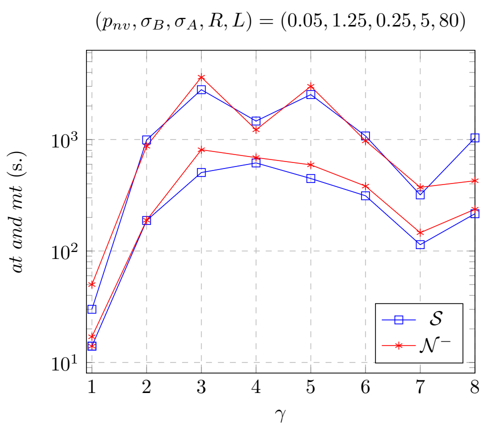

Figure 5: KC

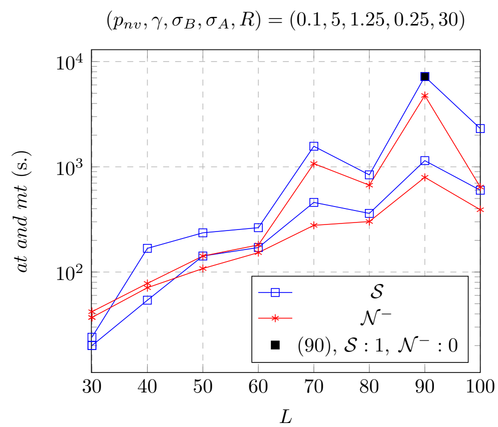

Figure 6: KC

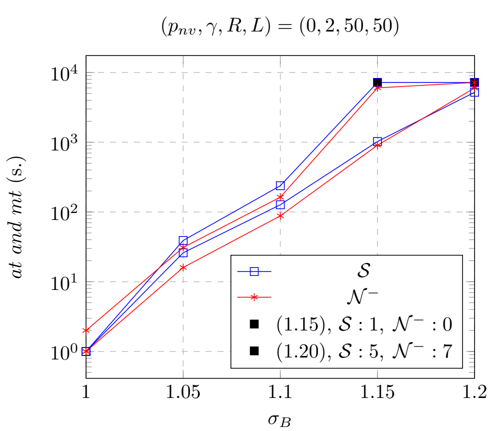

Figure 7: Km

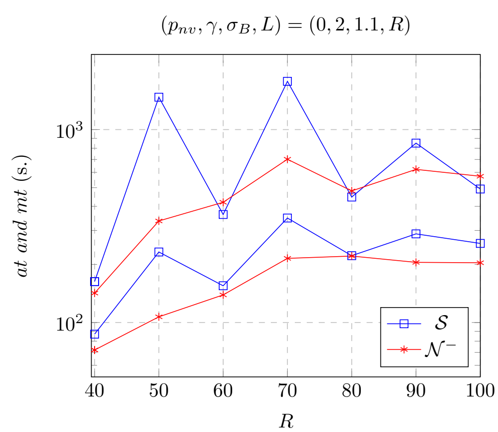

Figure 8: Km

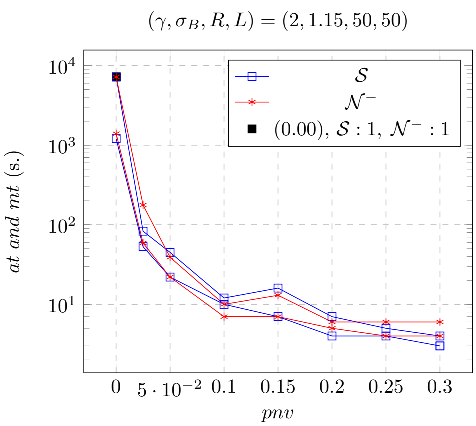

Figure 9: Km

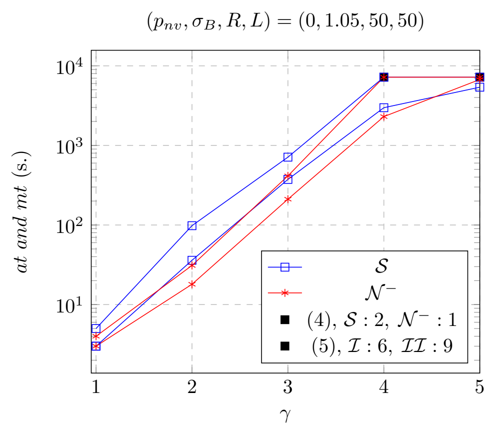

Figure 10: Km

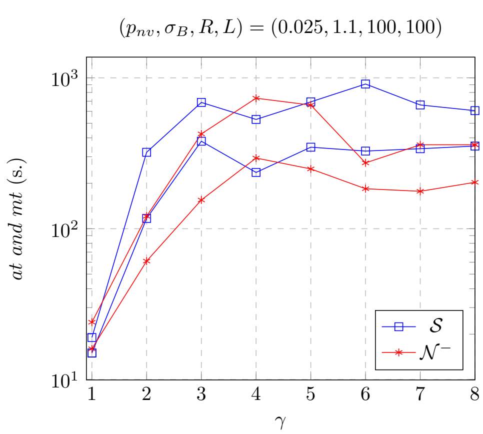

Figure 11: Km

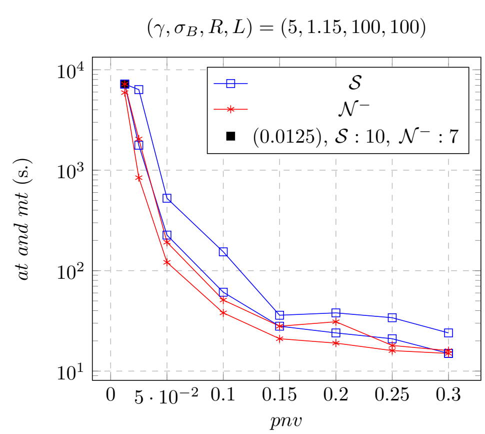

Figure 12: Km

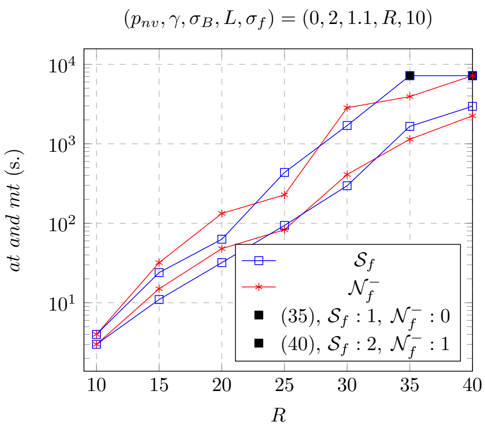

Figure 13: Km_f

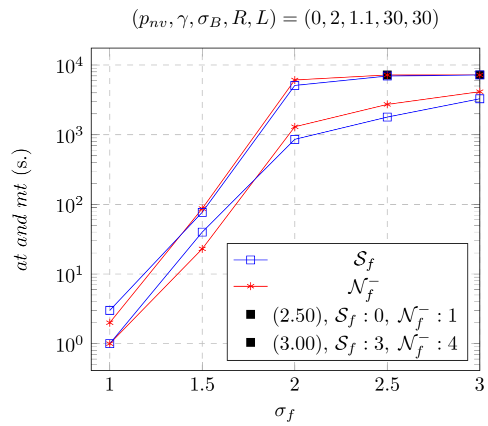

Figure 14: Km_f

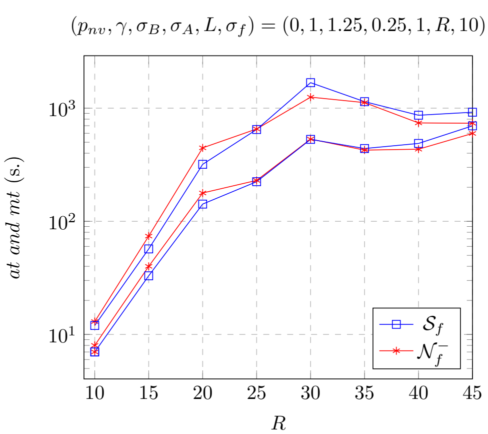

Figure 15: KC_f

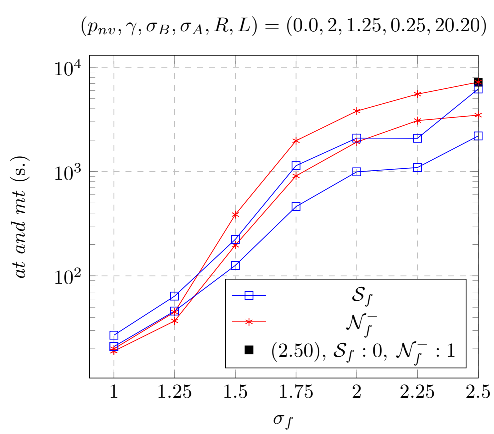

Figure 16: KC_f

[2] T. Szkaliczki, Solution methods for the multiple-choice knapsack problem and their applications, Mathematics 13 (7) (2025). doi:https://www.mdpi.com/2227-7390/13/7/1097.

[3] M. Goerigk, M. Khosravi, Robust combinatorial optimization problems under budgeted interdiction uncertainty, OR Spectrum (2024). doi:https://doi.org/10.1007/s00291-024-00772-0. URL `https://doi.org/10.1007/s00291-024-00772-0`

[4] M. Goerigk, S. Lendl, Robust combinatorial optimization with locally budgeted uncertainty, Open Journal of Mathenatical Optimization 2(3) (2021) 1–18. doi:https://doi.org/10.1007/s00291-024-00772-0.
URL `http://ojmo.centre-mersenne.org/item/OJMO-2021-2-A-30`

[5] A. A. Pessoa, M. Poss, M. C. Roboredo, L. Aizemberg, Solving bilevel combinatorial optimization as bilinear min-max optimization via a branch-and-cut algorithm, Anais do XLV Simpósio Brasileiro de Pesquisa Operacional (SBPO) (2013).

[6] A. Crema, Robust combinatorial optimization problems with knapsack constraints under interdiction uncertainty, Optimization Online (2024).

[7] B. L. Gorissen, İhsan Yanikoğlu, D. den Hertog, A practical guide to robust optimization, Omega 53 (2015) 124–137. doi:https://doi.org/10.1016/j.omega.2014.12.006.
URL `https://www.sciencedirect.com/science/article/pii/S0305048314001698`

[8] M. E. Pfetsch, A. Schmitt, A generic optimization framework for resilient systems, Optimization Methods and Software 38 (2) (2023) 356–385. doi:10.1080/10556788.2022.2142581.
URL `https://doi.org/10.1080/10556788.2022.2142581`

[9] D. Pisinger, Where are the hard knapsack problems?, Computers and Operations Research 32 (9) (2005) 2271–2284.

[10] E. Danna, Performance variability in mixed-integer programming, Presentation, Workshop on Mixed Interger Programming (MIP 2008), Columbia University, New York (2008).

[11] A. Lodi, A. Tramontani, Performance variability in mixed-integer programming, INFORMS TutORials in Operations Research (2013) 1–12doi:https://doi.org/10.1287/educ.2013.0112.

[12] M. Fischetti, A. Lodi, Local branching, Math. Program., Ser. B 98 (2003) 23–47. doi:https://doi.org/10.1007/s10107-003-0395-5.

[13] R. Bellman, Dynamic Programming, Press Princeton, Princeton, NJ, USA, 1957.