Min max (relative) set-regret combinatorial optimization

Alejandro Crema

October 2018

Alejandro Crema

Escuela de Computación, Facultad de Ciencias, Universidad Central de Venezuela, Caracas, Venezuela.
email: alejandro.crema@ciens.ucv.ve

Abstract

We consider combinatorial optimization problems with uncertainty in the cost vector. Recently a novel approach was developed to deal such uncertainties: instead of a single one robust solution, obtained by solving a min max problem, the authors consider a set of solutions obtained by solving a min max min problem. In this new approach the set of solutions is computed once and we can choose the best one in real time each time a cost vector occurs yielding better solutions compared to the min max approach. In this paper we extend the new approach by considering the absolute and relative deviation from the optimal values. Algorithms to solve the new min max (relative) set-regret problems are presented with a computational experience.

Keywords: Minmax regret, Robust Programming, Greedy algorithms, Multiparametric Programming.

1 Introduction

A few words about our notation: If R is an optimization problem then $v(R)$ is its optimal value. If we write $R(\boldsymbol{\theta}, \cdots, \boldsymbol{\gamma})$ is a problem in $(\mathbf{x}, \cdots, \mathbf{y}, \mathbf{H})$ that means that $\mathbf{x}, \cdots, \mathbf{y}$ are variable vectors and $\mathbf{H}$ is a variable set and $\boldsymbol{\theta}, \cdots, \boldsymbol{\gamma}$ are data vectors or data matrices that may change from one problem to another. The rest of the data for R are fixed and that must be clear in the context.

Data uncertainty appears in many optimization problems. In recent decades *Robust* (Aissi et al. 2009, Averbakh 2005, Candia-Véjar 2011, Kasperski and Zielinski 2016), *Stochastic* (Li and Liu 2016) and *Multiparametric* (Oberdieck et al. 2016) approaches have been developed to deal such uncertainties. In this paper we have used the Robust approach.

We consider combinatorial optimization problems in $(\mathbf{x})$ with interval uncertainty in the cost vector as follows:

$$(P(\mathbf{c}))\ \ \min\{\mathbf{c}^t\mathbf{x} : \mathbf{x} \in X\}$$

where $\mathbf{c} \in \Omega = \{\mathbf{c} : \mathbf{L} \leq \mathbf{c} \leq \mathbf{U}\}$, $\mathbf{L} \in \mathbb{R}^n$, $\mathbf{U} \in \mathbb{R}^n$ with $\mathbf{L} \neq \mathbf{0}$, $\mathbf{0} \leq \mathbf{L} \leq \mathbf{U}$, $X \subseteq \{0,1\}^n$, X is not an empty set, $\mathbf{c}^t\mathbf{x} > 0$ for all $\mathbf{c} \in \Omega$ and for all $\mathbf{x} \in X$

and $v(P(\mathbf{c})) > 0 \ \ \forall \mathbf{c} \in \Omega$.

A first approach of robust optimization is to find a solution that is optimal in the worst case, known as *robust solution*, by solving the next problem in $(\mathbf{x})$:

$$\min\{\max\{\mathbf{c}^t\mathbf{x} : \mathbf{c} \in \Omega\} : \mathbf{x} \in X\}$$

Recently a novel approach has been presented considering a set of k solutions instead of a single one (Buchheim and Kurtz 2016, Buchheim and Kurtz 2017). The problem in (H) to be solved is:

$$\min\{\max\{\min\{\mathbf{c}^t\mathbf{h} : \mathbf{h} \in H\} : \mathbf{c} \in \Omega\} : H \subseteq X, \ \ |H| = k\}$$

The idea behind this approach is to find a set of solutions and choose the best one each time a new scenario $\mathbf{c}$ appears.

As an application, imagine an emergency service designed based on the p-Medians problem (Boffey and Karkazis 1984). Each time that changes the current situation a new set of Medians could be computed. However, this may be a hard task. Even if the computational effort is not large an excessive number of solutions may be unacceptable for human users. Instead, in this new approach a set of solutions is computed once and then we can choose the best one in real time taken from a relatively small set of solutions yielding better performance compared to the min max approach.

An approach of robust optimization taking in account the absolute deviation from the optimal values is to find a solution with a minimun *regret* known as an *absolute robust solution*. The regret of $\mathbf{x}$ is equal to $\max\{\mathbf{c}^t\mathbf{x} - v(P(\mathbf{c})) : \mathbf{c} \in \Omega\}$. In this case the problem to be solved is the *min max regret* problem in $(\mathbf{x})$:

$$\min\{\max\{\mathbf{c}^t\mathbf{x} - v(P(\mathbf{c})) : \mathbf{c} \in \Omega\} : \mathbf{x} \in X\}$$

An approach of robust optimization taking in account the relative deviation from the optimal values is to find a solution with a minimun *relative regret* known as a *relative robust solution*. The relative regret of $\mathbf{x}$ is equal to $\max\{\frac{\mathbf{c}^t\mathbf{x} - v(P(\mathbf{c}))}{v(P(\mathbf{c}))} : \mathbf{c} \in \Omega\}$. In this case the problem to be solved is the *min max relative regret* problem in $(\mathbf{x})$:

$$\min\{\max\{\frac{\mathbf{c}^t\mathbf{x} - v(P(\mathbf{c}))}{v(P(\mathbf{c}))} : \mathbf{c} \in \Omega\} : \mathbf{x} \in X\}$$

In this paper we extend the novel approach presented above considering a set of k solutions and taking in account the absolute and relative deviations from the optimal values.

Let $H \subseteq X$. Let $Q(H)$ be a problem in $(\mathbf{c})$ defined as follows:

$$(Q(H))\ \ \max\{\min\{\mathbf{c}^t\mathbf{h} : \mathbf{h} \in H\} - v(P(\mathbf{c})) : \mathbf{c} \in \Omega\}$$

The *regret* of H is equal to $v(Q(H))$.

Let $Q_r(H)$ be a problem in $(\mathbf{c})$ defined as follows:

$$(Q_r(H))\ \ \max\{\frac{\min\{\mathbf{c}^t\mathbf{h} : \mathbf{h} \in H\} - v(P(\mathbf{c}))}{v(P(\mathbf{c}))} : \mathbf{c} \in \Omega\}$$

The *relative regret* of H is equal to $v(Q_r(H))$.

We are looking for the set with cardinality equal to k and with the minimun regret (minimun relative regret) by solving the *min max set-regret*(k) (MMSR(k)) problem (*min max relative set-regret*(k) (MMrelSR(k)) problem) in (H), defined as follows:

$$\text{(MMSR(k))}\ \ \min\{v(Q(H)) : H \subseteq X,\ \ |H| = k\}\} =$$
$$\min\{\max\{\min\{\mathbf{c}^t\mathbf{h} : \mathbf{h} \in H\} - v(P(\mathbf{c})) : \mathbf{c} \in \Omega\} : H \subseteq X,\ \ |H| = k\}$$

$$\text{(MMrelSR(k))}\ \ \min\{v(Q_r(H)) : H \subseteq X,\ \ |H| = k\}\} =$$
$$\min\{\max\{\frac{\{\min\{\mathbf{c}^t\mathbf{h} : \mathbf{h} \in H\} - v(P(\mathbf{c}))}{v(P(\mathbf{c}))} : \mathbf{c} \in \Omega\} : H \subseteq X,\ \ |H| = k\}$$

In section 2 we present an ϵ-optimal algorithm (with $\epsilon \geq 0$ a tolerance predefined) to solve the MMSR(k) problem that may be considered a generalization of a classical approach where $k = 1$ (Aissi et al. 2009). Solving the MMSR(k) problem may be a hard task and it may be useful to consider two greedy approaches to be presented in section 3.

In sections 4 we consider the MMrelSR(k) problem and we present an ϵ-optimal algorithm to solve it. Our presentation follow the same scheme that we used for the MMSR(k) problem and the proof of some lemmas are analogous. Thus, in the sake of simplicity and in order to save space we omit some proofs. Solving the MMrelSR(k) problem may be a hard task and it may be useful to consider two greedy approaches to be presented in section 5.

Computational results are presented in section 6 for the p-Medians problem, the Minimum Spanning Tree problem (Montemanni and Gambardella 2005A) , the Shortest Path problem (Montemanni and Gambardella 2005B) and the Set Covering problem (Caprara et al. 2000). Conclusions and further extensions are presented in section 7.

2 An algorithm to solve the MMSR(k) problem

Let $\mathbf{x} \in X$. The *best scenario* for $\mathbf{x}$ is defined as follows: $\mathbf{c}^+(\mathbf{x})_j = \mathbf{L}_j\mathbf{x}_j + \mathbf{U}_j(1-\mathbf{x}_j)$ for all j.

Note that $\mathbf{c}^+(\mathbf{x})^t\mathbf{x} = \mathbf{L}^t\mathbf{x} \ \ \forall \mathbf{x} \in X$ and $\mathbf{c}^t\mathbf{h} - \mathbf{c}^t\mathbf{x} \leq \mathbf{c}^+(\mathbf{x})^t\mathbf{h} - \mathbf{c}^+(\mathbf{x})^t\mathbf{x} = \mathbf{c}^+(\mathbf{x})^t\mathbf{h} - \mathbf{L}^t\mathbf{x} \ \ \forall \mathbf{h} \in X, \ \ \forall \mathbf{x} \in X$. Also, $\mathbf{c}^t\mathbf{x} \geq \mathbf{c}^+(\mathbf{x})^t\mathbf{x} = \mathbf{L}^t\mathbf{x} \ \ \forall \mathbf{x} \in X, \forall \mathbf{c} \in \Omega$.

Lemma 1 *$Q(H)$ may be rewritten as a problem in $(\sigma, \mathbf{x})$ as follows:*

$$(Q(H)) \ \ max\{\sigma - \mathbf{L}^t\mathbf{x} : \sigma \leq \mathbf{c}^+(\mathbf{x})^t\mathbf{h} \ \ \forall \mathbf{h} \in H, \ \ \mathbf{x} \in X\}$$

Proof:

$$v(Q(H)) = \max\{\min\{\mathbf{c}^t\mathbf{h} : \mathbf{h} \in H\} - v(P(\mathbf{c})) : \mathbf{c} \in \Omega\} \leq$$
$$\max\{\min\{\mathbf{c}^t\mathbf{h} : \mathbf{h} \in H\} - \mathbf{c}^t\mathbf{x} : \mathbf{c} \in \Omega, \ \ \mathbf{x} \in X\} \leq$$
$$\max\{\min\{\mathbf{c}^+(\mathbf{x})^t\mathbf{h} : \mathbf{h} \in H\} - \mathbf{c}^+(\mathbf{x})^t\mathbf{x} : \mathbf{x} \in X\} \leq v(Q(H))$$

and

$$\max\{\min\{\mathbf{c}^+(\mathbf{x})^t\mathbf{h} : \mathbf{h} \in H\} - \mathbf{c}^+(\mathbf{x})^t\mathbf{x} : \mathbf{x} \in X\} =$$
$$\max\{\sigma - \mathbf{L}^t\mathbf{x} : \sigma \leq \mathbf{c}^+(\mathbf{x})^t\mathbf{h} \ \ \forall \mathbf{h} \in H, \ \ \mathbf{x} \in X\} \ \ \bullet$$

Note that if $\mathbf{c}^*$ is an optimal solution for $Q(H)$ written as a problem in $(\mathbf{c})$ and $\mathbf{x}^*$ is an optimal solution for $P(\mathbf{c}^*)$ then $(\sigma^*, \mathbf{x}^*)$ is an optimal solution for $Q(H)$ written as a problem in $(\sigma, \mathbf{x})$ where $\sigma^* = \min\{\mathbf{c}^{*t}\mathbf{h} : \mathbf{h} \in H\}$.

Also, if $(\sigma^*, \mathbf{x}^*)$ is an optimal solution for $Q(H)$ written as a problem in $(\sigma, \mathbf{x})$ then $\mathbf{c}^* = \mathbf{c}^+(\mathbf{x}^*)$ is an optimal solution for $Q(H)$ written as a problem in $(\mathbf{c})$.

Lemma 2 *Let $Y \subseteq X$. Let $S(Y)$ be a problem in (σ, H) defined as follows:*

$(S(Y)) \ \ min\{\sigma : \sigma \geq \mathbf{c}^+(\mathbf{x})^t\mathbf{h} - \mathbf{L}^t\mathbf{x}$ *for some* $\mathbf{h} \in H \ \forall \mathbf{x} \in Y, \ \ H \subseteq X, \ \ |H| = k\}$

then $v(MMSR(k)) \geq v(S(Y))$

Proof:

$$v(\text{MMSR}(k)) = \min\{v(Q(H)) : H \subseteq X,\ \ |H| = k\} =$$
$$\min\{\max\{\min\{\mathbf{c}^t\mathbf{h} : \mathbf{h} \in H\} - v(P(\mathbf{c})) : \mathbf{c} \in \Omega\} : H \subseteq X,\ \ |H| = k\} =$$
$$\min\{\max\{\min\{\mathbf{c}^+(\mathbf{x})^t\mathbf{h} : \mathbf{h} \in H\} - \mathbf{L}^t\mathbf{x} : \mathbf{x} \in X\} : H \subseteq X,\ \ |H| = k\}\} =$$
$$\min\{\sigma : \sigma \geq \min\{\mathbf{c}^+(\mathbf{x})^t\mathbf{h} : \mathbf{h} \in H\} - \mathbf{L}^t\mathbf{x}\ \ \forall \mathbf{x} \in X,\ \ H \subseteq X,\ \ |H| = k\} \geq$$
$$\min\{\sigma : \sigma \geq \min\{\mathbf{c}^+(\mathbf{x})^t\mathbf{h} : \mathbf{h} \in H\} - \mathbf{L}^t\mathbf{x}\ \ \forall \mathbf{x} \in Y,\ \ H \subseteq X,\ \ |H| = k\} =$$
$$\min\{\sigma : \sigma \geq \mathbf{c}^+(\mathbf{x})^t\mathbf{h} - \mathbf{L}^t\mathbf{x} \text{ for some } \mathbf{h} \in H\ \forall \mathbf{x} \in Y,\ \ H \subseteq X,\ \ |H| = k\}\ \bullet$$

In order to solve $S(Y)$ we use the *Big-M* formulation. $S(Y)$ may be rewritten as a problem in $(\sigma, H, \mathbf{s}, \mathbf{z})$ as follows:

$$(S(Y))\ \ \min\{\sigma :$$
$$\sigma \geq s_{\mathbf{x}} - \mathbf{L}^t\mathbf{x}\ \forall \mathbf{x} \in Y,$$
$$\mathbf{s}_{\mathbf{x}} \geq \mathbf{c}^+(\mathbf{x})^t\mathbf{h}^j - (1 - \mathbf{z}_{(j,\mathbf{x})})M_{\mathbf{x}}\ \ \forall \mathbf{x} \in Y\ \ \forall j,$$
$$\sum_{j=1}^{k} \mathbf{z}_{(j,\mathbf{x})} = 1\ \ \forall \mathbf{x} \in Y,$$
$$\mathbf{h}^{(j)} \in X\ \ \forall j,\ \ \mathbf{z}_{(j,\mathbf{x})} \in \{0,1\}\ \ \forall \mathbf{x} \in Y,\ \ \forall j\}$$

where $H = \{\mathbf{h}^{(1)}, \cdots, \mathbf{h}^{(k)}\}$ and we use $M_{\mathbf{x}} = \max\{\mathbf{c}^+(\mathbf{x})^t\mathbf{w} : \mathbf{w} \in X\}$ as Big-M values with optimality guaranteed.

The S-Q algorithm to solve the MMSR(k) problem

Let $\epsilon \geq 0$. Let $H \subseteq X$ with $|H| = k$. Let $Y \subseteq X$. Let $UB = \infty$. Let $LB = 0$. The output is H with $v(Q(H)) - v(MMSR(k)) \leq \epsilon$.

1. Solve $Q(H)$. Let $(\sigma_H, \mathbf{y})$ be an optimal solution and let $UB = min\{UB, v(Q(H))\}$.
2. If $UB - LB \leq \epsilon$ Stop, otherwise let $Y = Y \cup \{\mathbf{y}\}$.
3. Solve $S(Y)$. Let $(\sigma_S, H, \mathbf{s}, \mathbf{z})$ be an optimal solution and let $LB = \sigma_S$.
4. If $UB - LB \leq \epsilon$ Stop, otherwise return to step 1.

Lemma 3 *Algorithm S-Q finds an ϵ-optimal solution for the MMSR(k) problem in a finite number of steps.*

Proof Let $(\sigma_S, H, \mathbf{s}, \mathbf{z})$ be an optimal solution for $S(Y)$ and let $(\sigma_H, \mathbf{y})$ be an optimal solution for $Q(H)$. If $\mathbf{y} \in Y$ then

$$\sigma_S \geq \mathbf{c}^+(\mathbf{y})^t\mathbf{h} - \mathbf{L}^t\mathbf{y} \text{ for some } \mathbf{h} \in H$$

and then

$$LB = \sigma_S \geq \min\{\mathbf{c}^+(\mathbf{y})^t\mathbf{h} : \mathbf{h} \in H\} - \mathbf{L}^t\mathbf{y} = v(Q(H)) \geq UB \geq$$
$$v(MMSR(k)) \geq v(S(Y)) = \sigma_S = LB$$

Since X is a finite set then $UB - LB \leq \epsilon$ in a finite number of steps •

Note that if $k = 1$ then the S-Q algorithm it turns to be a classic known algorithm to find an absolute robust solution (Aissi et al. 2009).

Solving the MMSR(k) problem may be a hard task and then we consider a greedy approaches to be presented in the next section.

3 Greedy aproaches for the MMSR(k) problem.

3.1 A greedy algorithm for the MMSR(k) problem based on *conditionated absolute robust solutions*

Let $H \subseteq X$ and let $T(H)$ be a problem in $(\mathbf{x})$ defined as follows:

$$(T(H)) \ \min\{v(Q(H \cup \{\mathbf{x}\}) : \mathbf{x} \in X\}$$

We say that $v(Q(H \cup \mathbf{x}))$ is the *regret of* $\mathbf{x}$ *conditionated by* H and if $\mathbf{x}^*$ is an optimal solution for $T(H)$ then we say that $\mathbf{x}^*$ is an *absolute robust solution conditionated by* H.

The T-greedy algorithm for the MMSR(k) problem

Let $H \subseteq X$ with $|H| = 1$.

1. Solve $T(H)$ and let $\mathbf{x}$ be an optimal solution.
2. If $v(T(H)) = 0$ Stop.
3. If $|H| = k$ then Stop, otherwise let $H = H \cup \{\mathbf{x}\}$ and return to step 1.

The T-greedy algorithm is designed in such a manner that the next solution to be included in H is an absolute robust solution conditionated by H. If $v(T(H)) = 0$ at step 2 then we do not need another solution even if $|H| < k$.

Note that the T-greedy algorithm may be seen as the first k steps of an algorithm to find an *optimal multiparametric solution for* P *relative to* Ω (Crema

2004).

$\hat{X}$ is an optimal multiparametrical solution for P relative to Ω if $\hat{X} \subseteq X$ and $v(Q(\hat{X})) = 0$. In that case $min\{\mathbf{c}^t\mathbf{x} : \mathbf{x} \in \hat{X}\} = v(P(\mathbf{c}))$ for all $\mathbf{c} \in \Omega$. If we use the T-greedy algorithm with $k = \infty$ then $v(T(H)) = 0$ at step 2 in a finite number of steps because of X is a finite set.

In order to solve $T(H)$ we present below an approach that may be seen, again, as a generalization of a standard approach to find an absolute robust solution.

Lemma 4 *Let $Y \subseteq X$. Let $\phi(\mathbf{y}) = min\{\mathbf{c}^+(\mathbf{y})^t\mathbf{h} : \mathbf{h} \in H\}\ \ \forall \mathbf{y} \in Y$*

Let $W(H, Y)$ be a problem in $(\sigma, \mathbf{x})$ defined as follows:

$$min\{\sigma : \sigma \geq \phi(\mathbf{y}) - \mathbf{L}^t\mathbf{y}\ \ \vee\ \ \sigma \geq \mathbf{c}^+(\mathbf{y})^t\mathbf{x} - \mathbf{L}^t\mathbf{y}\ \ \forall \mathbf{y} \in Y,\ \ \mathbf{x} \in X\}$$

then $v(T(H)) \geq v(W(H, Y))$

Proof:

$$\begin{gathered}
v(T(H)) = \min\{v(Q(H) \cup \mathbf{x}) : \mathbf{x} \in X\} = \\
\min\{\max\{\min\{\mathbf{c}^t\mathbf{h} : \mathbf{h} \in H \cup \{\mathbf{x}\}\} - v(P(\mathbf{c})) : \mathbf{c} \in \Omega\} : \mathbf{x} \in X\} = \\
\min\{\max\{\min\{\mathbf{c}^+(\mathbf{y})^t\mathbf{h} : \mathbf{h} \in H \cup \{\mathbf{x}\}\} - \mathbf{L}^t\mathbf{y} : \mathbf{y} \in X\} : \mathbf{x} \in X\} \geq \\
\min\ \ \{\sigma : \sigma \geq \min\{\mathbf{c}^+(\mathbf{y})^t\mathbf{h} - \mathbf{L}^t\mathbf{y} : \mathbf{h} \in H \cup \{\mathbf{x}\}\}\ \ \forall \mathbf{y} \in Y,\ \ \mathbf{x} \in X\} = \\
\min\{\sigma : \sigma \geq \phi(\mathbf{y}) - \mathbf{L}^t\mathbf{y}\ \ \vee\ \ \sigma \geq \mathbf{c}^+(\mathbf{y})^t\mathbf{x} - \mathbf{L}^t\mathbf{y}\ \ \forall \mathbf{y} \in Y,\ \ \mathbf{x} \in X\}\ \bullet
\end{gathered}$$

In order to solve $W(H, Y)$ we use the Big-M formulation. $W(H, Y)$ may be rewritten as a problem in $(\sigma, \mathbf{x}, \mathbf{s}, \mathbf{z})$ as follows:

$$\begin{gathered}
(W(H,Y))\ \ \min\{\sigma : \sigma \geq \mathbf{s_y} - \mathbf{L}^t\mathbf{y}, \\
\mathbf{s_y} \geq \phi(\mathbf{y})\mathbf{z_y}\ \ \forall \mathbf{y} \in Y, \\
\mathbf{s_y} \geq \mathbf{c}^+(\mathbf{y})^t\mathbf{x} - M_{\mathbf{y}}\mathbf{z_y}\ \ \forall \mathbf{y} \in Y, \\
\mathbf{x} \in X,\ \ \mathbf{z_y} \in \{0,1\}\ \ \forall \mathbf{y} \in Y\}
\end{gathered}$$

and we use $M_{\mathbf{y}} = \max\{\mathbf{c}^+(\mathbf{y})^t\mathbf{w} : \mathbf{w} \in X\}$ as Big-M values with optimality guaranteed.

W-Q algorithm to solve the T(H) problem

Let $\epsilon \geq 0$. Let $H \subseteq X$. Let $Y \subseteq X$. Let $\mathbf{x} \in X$. Let $UB = \infty$. Let $LB = 0$. The output is $\mathbf{x}$ with $v(Q(H \cup \{\mathbf{x}\})) - v(T(H)) \leq \epsilon$.

1. Solve $Q(H \cup \{\mathbf{x}\})$. Let $(\sigma_H, \mathbf{y})$ be an optimal solution and let $UB = min\{UB, v(Q(H \cup \{\mathbf{x}\}))$.
2. If $UB - LB \leq \epsilon$ Stop, otherwise let $Y = Y \cup \{\mathbf{y}\}$.
3. Solve $W(H, Y)$ and let $(\sigma_W, \mathbf{x}, \mathbf{s}, \mathbf{z})$ be an optimal solution. Let $LB = \sigma_W$.
4. If $UB - LB \leq \epsilon$ Stop, otherwise return to step 1.

Lemma 5 *Algorithm W-Q finds an ϵ-optimal solution for $T(H)$ in a finite number of steps.*

Let $(\sigma_W, \mathbf{x}, \mathbf{s}, \mathbf{z})$ be an optimal solution for $W(H, Y)$ and let $(\sigma_H, \mathbf{y})$ be an optimal solution for $Q(H \cup \{\mathbf{x}\})$. If $\mathbf{y} \in Y$ then

$$v(T(H)) \geq v(W(H, Y)) \geq \min\{\min\{\mathbf{c}^+(\mathbf{y})^t\mathbf{h} : \mathbf{h} \in H\}, \mathbf{c}^+(\mathbf{y})^t\mathbf{x}\} - \mathbf{L}^t\mathbf{y} =$$

$$v(Q(H \cup \{\mathbf{x}\})) \geq UB \geq v(T(H))$$

Since X is a finite set then $UB - LB \leq \epsilon$ in a finite number of steps •

3.2 A greedy algorithm for the MMSR(k) problem based on a simple multiparametric algorithm

The Q-greedy algorithm for the MMSR(k) problem

Let $H \subseteq X$ with $|H| = 1$.

1. Solve $Q(H)$. Let $(\sigma, \mathbf{x})$ be an optimal solution.
2. If $v(Q(H)) = 0$ Stop.
3. If $|H| = k$ Stop, otherwise let $H = H \cup \{\mathbf{x}\}$ and return to step 1.

The Q-greedy algorithm is defined as the first k steps of a simple multiparametric algorithm (Crema 2000). In each step we are looking for the worst scenario for H. If $(\sigma^*, \mathbf{x}^*)$ is an optimal solution for $Q(H)$ then $\mathbf{c}^+(\mathbf{x}^*)$ is the worst scenario for H and $\mathbf{x}^*$ is choosed to be included in H.

For the *same* H if x^1 is the solution generated by the Q-greedy algorithm and x^2 is the solution generated by the T-greedy algorithm then $v(Q(H \cup \mathbf{x^{(1)}})) \geq v(Q(H \cup \mathbf{x^{(2)}}))$. Thus, it can be expected, but it is not safe from the theoretical point of view, that the regret achieved for the last H using the T-greedy algorithm will be better that using the Q-greedy algorithm. The computational results presented in section 6 confirm that expectation. The trade off between the quality of the last H and the computational effort must be considered by the decision maker.

4 An algorithm to solve the MMrelSR(k) problem

Lemma 6 *$Q_r(H)$ may be rewritten as a problem in $(\mathbf{x})$ as follows:*

$$max\{\frac{min\{\mathbf{c}^+(\mathbf{x})^t\mathbf{h} : \mathbf{h} \in H\} - \mathbf{c}^+(\mathbf{x})^t\mathbf{x}}{\mathbf{c}^+(\mathbf{x})^t\mathbf{x}} : \mathbf{x} \in X\}$$

Proof:

Let $\mathbf{c}^*$ be an optimal solution for $Q_r(H)$ and let $\mathbf{x}^*$ be an optimal solution for $P(\mathbf{c}^*)$. Then we have:

$$v(Q_r(H)) = \frac{\min\{\mathbf{c}^{*t}\mathbf{h} : \mathbf{h} \in H\} - \mathbf{c}^{*t}\mathbf{x}^*}{\mathbf{c}^{*t}\mathbf{x}^*} \leq$$
$$\max\{\frac{\min\{\mathbf{c}^t\mathbf{h} : \mathbf{h} \in H\} - \mathbf{c}^t\mathbf{x}}{\mathbf{c}^t\mathbf{x}} : \mathbf{c} \in \Omega,\ \ \mathbf{x} \in X\}$$

Since $\mathbf{c}^t\mathbf{x} \geq v(P(\mathbf{c}))$ for all $\mathbf{c} \in \Omega$ and $\forall \mathbf{x} \in X$ we have that

$$\max\{\frac{\min\{\mathbf{c}^t\mathbf{h} : \mathbf{h} \in H\} - \mathbf{c}^t\mathbf{x}}{\mathbf{c}^t\mathbf{x}} : \mathbf{c} \in \Omega,\ \ \mathbf{x} \in X\} \leq$$
$$\max\{\frac{\min\{\mathbf{c}^t\mathbf{h} : \mathbf{h} \in H\} - v(P(\mathbf{c}))}{v(P(\mathbf{c}))} : \mathbf{c} \in \Omega,\ \ \mathbf{x} \in X\}$$

Therefore,

$$v(Q_r(H)) = \max\{\frac{\min\{\mathbf{c}^t\mathbf{h} : \mathbf{h} \in H\} - \mathbf{c}^t\mathbf{x}}{\mathbf{c}^t\mathbf{x}} : \mathbf{c} \in \Omega,\ \ \mathbf{x} \in X\}$$

Because of $\mathbf{c}^t\mathbf{h} - \mathbf{c}^t\mathbf{x} \leq \mathbf{c}^+(\mathbf{x})^t\mathbf{h} - \mathbf{c}^+(\mathbf{x})^t\mathbf{x}$ and $\mathbf{c}^t\mathbf{x} \geq \mathbf{c}^+(\mathbf{x})^t\mathbf{x}\ \ \forall \mathbf{x} \in X,\ \ \forall \mathbf{c} \in \Omega\ \ \forall \mathbf{h} \in X$ we have:

$$\max\{\frac{\min\{\mathbf{c}^t\mathbf{h} : \mathbf{h} \in H\} - \mathbf{c}^t\mathbf{x}}{\mathbf{c}^t\mathbf{x}} : \mathbf{c} \in \Omega,\ \ \mathbf{x} \in X\} \leq$$
$$\max\{\frac{\min\{\mathbf{c}^+(\mathbf{x})^t\mathbf{h} : \mathbf{h} \in H\} - \mathbf{c}^+(\mathbf{x})^t\mathbf{x}}{\mathbf{c}^+(\mathbf{x})^t\mathbf{x}} : \mathbf{x} \in X\} \leq$$
$$\max\{\frac{\min\{\mathbf{c}^t\mathbf{h} : \mathbf{h} \in H\} - \mathbf{c}^t\mathbf{x}}{\mathbf{c}^t\mathbf{x}} : \mathbf{c} \in \Omega,\ \ \mathbf{x} \in X\}\ \bullet$$

Note that if $\mathbf{c}^*$ is an optimal solution for $Q_r(H)$ written as a problem in $(\mathbf{c})$ and $\mathbf{x}^*$ is an optimal solution for $P(\mathbf{c}^*)$ then $\mathbf{x}^*$ is an optimal solution for $Q_r(H)$ written as a problem in $(\mathbf{x})$

Also, if $\mathbf{x}^*$ is an optimal solution for $Q_r(H)$ written as a problem in $(\mathbf{x})$ then $\mathbf{c}^* = \mathbf{c}^+(\mathbf{x}^*)$ is an optimal solution for $Q_r(H)$ written as a problem in $(\mathbf{c})$.

Let $\mu \geq 0$ and let $R_r(\mu, H)$ a problem in $(\mathbf{x})$ defined as:

$$(R_r(\mu, H)) \ \max\{\min\{\mathbf{c}^+(\mathbf{x})^t\mathbf{h} : \mathbf{h} \in H\} - \mu\mathbf{c}^+(\mathbf{x})^t\mathbf{x} : \mathbf{x} \in X\}$$

$Q_r(H)$ is a combinatorial optimization problem with a rational objective function (Meggido 1979). Hence, we know some basic properties for $R_r(\mu, H)$ and $Q_r(H)$:

$v(R_r(\mu, H)) = 0$ if and only if $\mu = v(R_r(\mu, H)) = v(Q_r(H)) + 1$. If $v(R_r(\mu, H)) = 0$ and $\mathbf{x}^*$ is an optimal solution for $R_r(\mu, H)$ then $\mathbf{x}^*$ is an optimal solution for $Q_r(H)$. If $v(R_r(\mu, H)) = 0$ then $\mu \geq 1$.

Therefore, in order to solve $Q_r(H)$ we may use a very simple algorithm to find μ with $v(R_r(\mu, H)) = 0$.

In order to solve $R_r(\mu, H)$ we may rewritten it as a problem in $(\sigma, \mathbf{x})$ as follows:

$$\max\{\sigma - \mu\mathbf{L}^t\mathbf{x} : \sigma \leq \mathbf{c}^+(\mathbf{x})^t\mathbf{h} \ \ \forall\mathbf{h} \in H, \ \ \mathbf{x} \in X\}$$

Algorithm to find μ with $v(R_r(\mu, H)) = 0$

Let $\mu = 1$. Let $H \subseteq X$ with $|H| = k$. The output is μ with $v(Q_r(\mu, H)) = 0$ and $\mathbf{x}$ optimal for $Q_r(H)$.

1. Solve $R_r(\mu, H)$. Let $\mathbf{x}$ be an optimal solution.
2. If $v(R_r(\mu, H)) = 0$ Stop.
3. Let $\mu = \frac{min\{\mathbf{c}^+(\mathbf{x})^t\mathbf{h}:\mathbf{h}\in H\}}{\mathbf{c}^+(\mathbf{x})^t\mathbf{x}}$ and return to step 1.

The validity of the algorithm follows from the fact that $v(Q_r(\mu, H))$ is a piecewise linear and decreasing convex function in μ.

Lemma 7 *Let $Y \subseteq X$. Let $S_r(Y)$ be a problem in (σ, H) defined as follows:*

$$(S_r(Y)) \ \ min\{\sigma : \mathbf{L}^t\mathbf{x} \ \sigma \geq \mathbf{c}^+(\mathbf{x})^t\mathbf{h} - \mathbf{L}^t\mathbf{x} \text{ for some } \mathbf{h} \in H \ \forall\mathbf{x} \in Y, \ \ H \subseteq X, \ \ |H| = k\}$$

then $v(MMrelSR(k)) \geq v(S_r(Y))$

We omit the proof because of it is analogous to the proof of lemma 2.

In order to solve $S_r(Y)$ we use the Big-M formulation. $S_r(Y)$ may be rewritten as a problem in $(\sigma, H, \mathbf{s}, \mathbf{z})$ as follows:

$$(S_r(Y))\ \min\{\sigma :$$
$$\mathbf{L}^t\mathbf{x}\ \sigma \geq s_{\mathbf{x}} - \mathbf{L}^t\mathbf{x}\ \forall \mathbf{x} \in Y,\ \ \forall j,$$
$$\mathbf{s}_{\mathbf{x}} \geq \mathbf{c}^+(\mathbf{x})^t\mathbf{h}^j - (1 - \mathbf{z}_{(j,\mathbf{x})})M_{\mathbf{x}}\ \ \forall \mathbf{x} \in Y,$$
$$\sum_{j=1}^{k} \mathbf{z}_{(j,\mathbf{x})} = 1\ \ \forall \mathbf{x} \in Y,$$
$$\mathbf{h}^{(j)} \in X\ \ \forall j,\ \ \mathbf{z}_{(j,\mathbf{x})} \in \{0,1\}\ \ \forall \mathbf{x} \in Y,\ \ \forall j\}$$

where $H = \{\mathbf{h}^{(1)}, \cdots, \mathbf{h}^{(k)}\}$ and we use $M_{\mathbf{x}} = \max\{\mathbf{c}^+(\mathbf{x})^t\mathbf{w} : \mathbf{w} \in X\}$ as Big-M values with optimality guaranteed.

Sr-Qr algorithm to solve the MMrelSR(k) problem

Let $\epsilon \geq 0$. Let $H \subseteq X$ with $|H| = k$. Let $Y \subseteq X$. Let $UB = \infty$. Let $LB = 0$. The output is H with $v(Q_r(H)) - v(MMrelSR(k)) \leq \epsilon$.

1. Solve $Q_r(H)$. Let $\mathbf{y}$ be an optimal solution and let $UB = min\{UB, v(Q_r(H))\}$.
2. If $UB - LB \leq \epsilon$ Stop, otherwise let $Y = Y \cup \{\mathbf{y}\}$.
3. Solve $S_r(Y)$. Let $(\sigma_{S_r}, H, \mathbf{s}, \mathbf{z})$ be an optimal solution and let $LB = \sigma_{S_r}$.
4. If $UB - LB \leq \epsilon$ Stop, otherwise return to step 1

Lemma 8 *Algorithm Sr-Qr finds an ϵ-optimal solution for MMrelSR(k) in a finite number of steps.*

We omit the proof because of it is analogous to the proof of lemma 3.

5 Greedy approaches for the MMrelSR(k) problem

5.1 A greedy algorithm for the MMrelSR(k) problem based on *conditionated relative robust solutions*

Let $H \subseteq X$ and let $T_r(H)$ be a problem in $\mathbf{x}$ defined as follows:

$$(T_r(H)) : \min\ \ v(Q_r(H \cup \{\mathbf{x}\})\ \ s.t.\ \ \mathbf{x} \in X$$

We say that $v(Q_r(H \cup \mathbf{x}))$ is the *relative regret of* $\mathbf{x}$ *conditionated by* H and if $\mathbf{x}^*$ is an optimal solution for $T_r(H)$ then we say that $\mathbf{x}^*$ is a *relative robust*

solution conditionated by H

The Tr-greedy algorithm for the MMrelSR(k) problem

Let $H \subseteq X$ with $|H| = 1$.

1. Solve $T_r(H)$ and let $\mathbf{x}$ be an optimal solution.
2. If $v(T_r(H)) = 0$ Stop.
3. If $|H| = k$ then Stop, otherwise Let $H = H \cup \{\mathbf{x}\}$ and return to step 1.

The Tr-greedy algorithm is designed in such a manner that the next solution to be included in H is a relative robust solution conditionated by H. If $v(T_r(H)) = 0$ at step 3 then we do not need another solution even if $|H| < k$.

In order to solve $T_r(H)$ we present below an approach that is analogous to the approch presented to solve $T(H)$.

Lemma 9 *Let $Y \subseteq X$. Let $\phi(\mathbf{y}) = min\{\mathbf{c}^+(\mathbf{y})^t\mathbf{h} : \mathbf{h} \in H\}$ $\forall \mathbf{y} \in Y$*

Let $W_r(H, Y)$ be a problem in $(\sigma, \mathbf{x})$ defined as follows:

$$min\{\sigma : \mathbf{L}^t\mathbf{y}\sigma \geq \phi(\mathbf{y}) - \mathbf{L}^t\mathbf{y} \quad \vee \quad \mathbf{L}^t\mathbf{y}\sigma \geq \mathbf{c}^+(\mathbf{y})^t\mathbf{x} - \mathbf{L}^t\mathbf{y} \ \ \forall \mathbf{y} \in Y, \ \ \mathbf{x} \in X\}$$

then $v(T_r(H)) \geq v(W_r(H, Y))$

We omit the proof because of it is analogous to the proof of lemma 4.

In order to solve $W_r(H, Y)$ we use the Big-M formulation. $W_r(H, Y)$ may be rewritten as a problem in $(\sigma, \mathbf{x}, \mathbf{s}, \mathbf{z})$ as follows:

$$\begin{aligned}(W_r(H,Y)) \ \ \min\{\sigma : \mathbf{L}^t\mathbf{y}\ \sigma \geq \mathbf{s_y} - \mathbf{L}^t\mathbf{y},\\ \mathbf{s_y} \geq \phi(\mathbf{y})\mathbf{z_y} \ \ \forall \mathbf{y} \in Y,\\ \mathbf{s_y} \geq \mathbf{c}^+(\mathbf{y})^t\mathbf{x} - M_\mathbf{y}\mathbf{z_y} \ \ \forall \mathbf{y} \in Y,\\ \mathbf{x} \in X, \ \ \mathbf{z_y} \in \{0,1\} \ \ \forall \mathbf{y} \in Y\}\end{aligned}$$

and we use $M_\mathbf{y} = \max\{\mathbf{c}^+(\mathbf{y})^t\mathbf{x} : x \in X\}$ as Big-M values with optimality guaranteed.

Wr-Qr algorithm to solve $T_r(H)$

Let $\epsilon \geq 0$. Let $H \subseteq X$. Let $Y \subseteq X$. Let $\mathbf{x} \in X$. Let $UB = \infty$. Let $LB = 0$. The output is $\mathbf{x}$ with $v(Q_r(H \cup \{\mathbf{x}\})) - v(T_r(H)) \leq \epsilon$.

1. Solve $Q_r(H \cup \{\mathbf{x}\})$. Let $(\mathbf{y})$ be an optimal solution and let $UB = min\{UB, v(Q_r(H \cup \{\mathbf{x}\}))$.
2. If $UB - LB \leq \epsilon$ Stop, otherwise let $Y = Y \cup \{\mathbf{y}\}$.
3. Solve $W_r(H, Y)$ and let $(\sigma_W, \mathbf{x}, \mathbf{s}, \mathbf{z})$ be an optimal solution. Let $LB = \sigma_W$.
4. If $UB - LB \leq \epsilon$ Stop, otherwise return to step 1.

Lemma 10 *Algorithm Wr-Qr finds an ϵ-optimal solution for $T_r(H)$ in a finite number of steps.*

We omit the proof because of it is analogous to the proof of lemma 5.

5.2 A greedy algorithm for the MMrelSR(k) problem based on a simple multiparametric algorithm

The Qr-greedy algorithm

Let $H \subseteq X$ with $|H| = 1$.

1. Solve $Q_r(H)$. Let $\mathbf{x}$ be an optimal solution.
2. If $v(Q_r(H)) = 0$ Stop.
3. If $|H| = k$ Stop, otherwise let $H = H \cup \{\mathbf{x}\}$ and return to step 1.

For the *same* H if x^1 is the solution generated by the Qr-greedy algorithm and x^2 is the solution generated by the Tr-greedy algorithm then $v(Q_r(H \cup \mathbf{x^{(1)}})) \geq v(Q_r(H \cup \mathbf{x^{(2)}}))$. Thus, it can be expected, but it is not safe from the theoretical point of view, that the regret achieved with the last H using the Tr-greedy algorithm will be better that using the Qr-greedy algorithm. The trade off between the quality of the last H and the computational effort must be considered by the decision maker.

6 Computational results

6.1 Computer environment and the notation.

Our algorithms have been performed on a personal computer as follows:

Intel(R)CoreTM2Duo CPU, T5900, with 4.00 GB Ram, 2.20 GHz and Windows 8.1 Operating System. All the instances have been processed through the commercial ILOG-Cplex 12.4 (http://ibm-ilog-cplex-optimization-studio-acade.software.informer.com/12.4/) from a MATLAB code by using the branch and cut algorithm with the default parameters with the exception of the relative

tolerance parameter (**mipgap**) that is set to zero.

The information to be presented is not the same for all tables but in order to save space and simplify the exposition we present all the values reported:

1. prob: an index to the problem,
2. t1: the time in seconds to find $h^{(1)}$, an absolute robust solution, by using the S-Q algotithm with $k = 1$ beginning with $Y = \emptyset$ and $H = \{\mathbf{x_L}\}$ where $\mathbf{x_L}$ is an optimal solution for $P(\mathbf{L})$,
3. tT(k): the time in seconds to generate H where $2 \leq |H| \leq k$ by using the T-greedy algorithm beginning with $H = \{h^{(1)}\}$ and Y obtained previously to compute $h^{(1)}$. In order to solve $T(H)$ by using the W-Q algorithm the initial $\mathbf{x}$ is an optimal solution for $Q(H)$.
4. tQ(k): the time in seconds to generate H where $2 \leq |H| \leq k$ by using the Q-greedy algorithm beginning with $H = \{h^{(1)}\}$
5. t(k): the time in seconds to generate an optimal H where $|H| = k$ by using the S-Q algorithm (with $k \in \{2, 3\}$). If $k = 2$ beginning with $H = \{h^{(1)}, h^{(2)}\}$ where $h^{(2)}$ was generated by using the T-greedy algorithm and Y obtained previously to compute H. If $k = 3$ beginning with $H = \{h^1, h^2, h^3\}$ where $(h^{(1)}, h^{(2)})$ is an optimal solution for $S^{(2)}$ obtained using the S-Q algorithm with $k = 2$ and $h^{(3)}$ was obtained by using the T-greedy algorithm beginning from $H = \{h^{(1)}, h^{(2)}\}$ and Y obtained previously to compute $(h^{(1)}, h^{(2)})$,
6. R(k): the regret of an optimal H with $|H| = k$,
7. RT(k): the regret of H with $|H| = k$ obtained by using the T-greedy algorithm.
8. RQ(k): the regret of H with $|H| = k$ obtained by using the Q-greedy algorithm.
9. tU(2): the time in seconds to find a set H with regret equal to RT(2) by using the S-Q algorithm beginning without the information generated with the T-greedy algorithm. We use $Y = \emptyset$ and $H = \{\mathbf{x_L}, \mathbf{x_L}\}$,
10. to(2): the time in seconds to verify the optimality by using the S-Q algorithm beginning without the information generated with the T-greedy algorithm,
11. $red(*, k) = 100(\frac{R(1)-R*(k)}{R(1)})$ where $* \in \{Q, T\}$. $red(*, k)$ is the reduction porcentage relative to $R(1)$ if we use H (with $|H| = k$) generated by using the *-greedy algorithm.
12. t1r, tTr(k), tQr(k), $red(T_r, k)$ and $red(Q_r, k)$ are analogous values when the relative case is being considered.

6.2 Data generation

At the present time we have computational results for the p-Median (p-M) problem, the Minimun Spanning Tree (MST) problem, the Shortest Path (SP) problem and the Set Covering (SC) problem .

6.2.1 Data for the p-Medians problem

Let $J = \{1, \cdots, s\}$ a set of demand locations. Each demand location is a candidate to be a service location (a median). Let $\mathbf{l}, \mathbf{u}$ be known matrices such that $\mathbf{0} \leq \mathbf{l} \leq \mathbf{u}$. If the demand location j is assigned to the median i the cost belongs to $[\mathbf{l}_{ij}, \mathbf{u}_{ij}]$ $(i \in J)$ $(j \in J)$. Let p the number of medians to be selected.

Let $\mathbf{x}_i \in \{0,1\}$ $(i \in J)$ with $\mathbf{x}_i = 1$ if and only if the demand location i is selected to be a median.

Let $\mathbf{x}_{s(i-1)s+j} \in \{0,1\}$ $(i \in J)(j \in J)$ with $\mathbf{x}_{s+(i-1)s+j} = 1$ if and only if the demand location j is assigned to a median located at i.

Let $n = s + s^2$. Let $\mathbf{L} \in \mathbb{R}^n$ and $\mathbf{U} \in \mathbb{R}^n$ defined as $\mathbf{L}_i = \mathbf{U}_i = 0 \ \ \forall i \in J$ and $\mathbf{L}_{s+(i-1)s+j} = \mathbf{l}_{ij}, \ \ \mathbf{U}_{s+(i-1)s+j} = \mathbf{u}_{ij} \ \ \forall i \in J \ \ \forall j \in J$.

Let $X \subseteq \{0,1\}^n$ defined as follows:

$$X = \{\mathbf{x} : \sum_{i=1}^{s} \mathbf{x}_i = p, \ \ \sum_{i=1}^{s} \mathbf{x}_{s+(i-1)s+j} = 1 \ \ \forall j \in J,$$
$$\mathbf{x}_{s+(i-1)s+j} \leq \mathbf{x}_i \ \ \forall i \in J \ \ \forall j \in J,$$
$$\mathbf{x} \in \{0,1\}^n\}$$

Let $\mathbf{c} \in \mathbb{R}^n$ with: $\mathbf{c} \in [\mathbf{L}, \mathbf{U}]$. The p-medians problem with cost $\mathbf{c}$ is a problem in $\mathbf{x}$ defined as follows:

$$(P(\mathbf{c})) \ \ \min \ \ \{\mathbf{c}^t\mathbf{x} : \mathbf{x} \in X\}$$

Because X was written as a 0-1-Integer Programming model then $P((\mathbf{c}))$, $Q(H)$, $S(Y)$, $W(H,Y)$, $Q_r(H)$, $Q_r(\mu, H)$, $S_r(Y)$, and $W_r(H,Y)$ may be solved by using CPLEX.

The data were generated at random as follows: let $(\mathbf{x1}_j, \mathbf{x2}_j)$ be the location j taken from $U((0,100) \times (0,100))$, let $\mathbf{D}_j$ be the demand of location j taken from $U(0,100)$. Let $\mathbf{d}_{ij}$ be the distance from location i until location j computed as $\mathbf{d}_{ij} = |\mathbf{x1}_i - \mathbf{x1}_j| + |\mathbf{x2}_i - \mathbf{x2}_j|$. Let $\gamma > 0$. Let $\mathbf{l}_{ij} = \mathbf{d}_{ij}\mathbf{D}_j$ and let $\mathbf{u}_{ij} = (1 + r_{ij}\gamma)\mathbf{l}_{ij}$ where r_{ij} is taken from $\beta(a,b)$. For some problems the definitive data were obtained by rounding down the data generated.

6.2.2 Data for the Set Covering problem

Let A be a 0-1 $m \times n$ matrix and let $\mathbf{L}$ and $\mathbf{U}$ known vectors in $\mathbb{R}^n$ with $\mathbf{0} \leq \mathbf{L} \leq \mathbf{U}$. The cost of column j is $\mathbf{c}_j$ with $\mathbf{L} \leq \mathbf{c} \leq \mathbf{U}$. We say that a column j covers a row i if $A_{ij} = 1$. We are looking for the subset of columns with the minimun cost such that each row is covered by at least one column. Let $\mathbf{x}_j \in \{0,1\}$ with $\mathbf{x}_j = 1$ if and only if the column j is selected.

Let $X \subseteq \{0,1\}^n$ defined as follows:

$$X = \{\mathbf{x} : \sum_{j=1}^{n} A_{ij}\mathbf{x}_j \geq 1 \ \ \forall i, \ \ \mathbf{x} \in \{0,1\}^n\}$$

The Set Covering problem with cost $\mathbf{c}$ is a problem in $\mathbf{x}$ defined as follows:

$$(P(\mathbf{c})) \ \ \min \ \ \{\mathbf{c}^t\mathbf{x} : \mathbf{x} \in X\}$$

Because X was written as a 0-1-Integer Programming model then $P((\mathbf{c}))$, $Q(H)$, $S(Y)$, $W(H,Y)$, $Q_r(H)$, $Q_r(\mu,H)$, $S_r(Y)$, and $W_r(H,Y)$ may be solved by using CPLEX.

The data were generated at random as follows: let $(\mathbf{x1}_i, \mathbf{x2}_i)$ be the location (row) i taken from $U((0,100) \times (0,100))$, let $\mathbf{d}_{ij}$ be the distance from location i until location j computed as $\mathbf{d}_{ij} = |\mathbf{x1}_i - \mathbf{x1}_j| + |\mathbf{x2}_i - \mathbf{x2}_j|$. We say that the location (column) j may cover the nearest K locations (rows) with the cost $\mathbf{L}_j$ equal to the sum of the distances of the locations covered. Let $\gamma > 0$. Let $\mathbf{U}_j = (1 + r_j\gamma)\mathbf{L}_j$ where r_j is taken from $U(0,1)$.

6.2.3 Data for the Shortest Path problem

Consider a directed graph $G = (V,E)$. Let $\hat{v} \in V$ and $\hat{w} \in V$. We are looking for the path from $\hat{v}$ until $\hat{w}$ with the minimun cost.

For each $v \in V$ let $E^+(v) = \{(v,w) : (v,w) \in E\}$ and let $E^-(v) = \{(w,v) : (w,v) \in E\}$.

Let $\mathbf{x} \in \{0,1\}^{|E|}$ with $x_e = 1$ is the edge e is selected.

Let $X \subseteq \{0,1\}^{|E|}$ defined as follows:

$$X = \{\mathbf{x} :$$

$$\sum_{e:e\in E^+(\hat{v})} x_e = 1,$$

$$\sum_{e:e\in E^-(v)} x_e - \sum_{e\in E^+(v)} x_e = 0 \ \ \forall v \in V \ \text{ with } \ v \notin \{\hat{v},\hat{w}\},$$

$$\sum_{e:e\in E^-(\hat{w})} x_e = 1,$$

$$\mathbf{x} \in \{0,1\}^{|E|}\}$$

Let $\mathbf{L}$ and $\mathbf{U}$ known vectors in $\mathbb{R}^{|E|}$ with $\mathbf{0} \leq \mathbf{L} \leq \mathbf{U}$. Let $\mathbf{c} \in \mathbb{R}^n$ with: $\mathbf{L} \leq \mathbf{c} \leq \mathbf{U}$. The Shortest Path problem with cost $\mathbf{c}$ is a problem in $\mathbf{x}$ defined as follows:

$$(P(\mathbf{c}))\ \min\ \{\mathbf{c}^t\mathbf{x} : \mathbf{x} \in X\}$$

Because X was written as a 0-1-Integer Programming model then $P((\mathbf{c}))$, $Q(H)$, $S(Y)$, $W(H,Y)$, $Q_r(H)$, $Q_r(\mu,H)$, $S_r(Y)$, and $W_r(H,Y)$ may be solved by using CPLEX.

We use $G = (V,E)$ with the following simple mesh topology: node $(0,1)$ is connected with nodes $(1,1),\cdots,(1,n)$. Node (i,k) is connected with nodes $(i+1,1),\cdots,(i+1,n)$ with $i = 1,\cdots,m-1$. Nodes $(m,1),\cdots,(m,n)$ are connected with node $(m+1,1)$. We have $|V| = 2+nm$ and $|E| = 2n+n^2(m-1)$. We use $\hat{v} = (0,1)$ and $\hat{w} = (m+1,1)$. Every path from $\hat{v}$ until $\hat{w}$ has $m+1$ edges.

Let $\gamma > 0$. The data were generated as follows: for each $e \in E$ we generate $\mathbf{L}_e$ from $U(0,1000)$. Let $\gamma > 0$. Let $\mathbf{U}_e = (1+r_e\gamma)\mathbf{L}_e$ with r_e taken from $U(0,1)$.

6.2.4 Data for the Minimun Spanning Tree problem

Let $G = (V,E)$ a conected undirected graph with $|E| = n$. Let $\mathbf{L} \in \mathbb{R}^n$ and $\mathbf{U} \in \mathbb{R}^n$ with $\mathbf{0} \leq \mathbf{L} \leq \mathbf{U}$. Each edge e has cost $\mathbf{c}_e \in [\mathbf{L}_e, \mathbf{U}_e]$. If $T = (V,\hat{E})$ with $\hat{E} \subseteq E$ and T is a tree with $|\hat{E}| = |V|-1$ then T is a spanning tree of G. As usual let $\mathbf{x}_e \in \{0,1\}$ with $\mathbf{x}_e = 1$ if $e \in \hat{E}$ and $\mathbf{x}_e = 0$ otherwise. The Minimun Spamnning Tree (MST) problem with cost $\mathbf{c}$ is a problem in $\mathbf{x}$ is defined as follows:

$$(P(\mathbf{c}))\ \min\ \{\mathbf{c}^t\mathbf{x} : \mathbf{x} \in X\}$$

where $X \subseteq \{0,1\}^n$ and $\mathbf{x} \in X$ if and only $(V, E(\mathbf{x}))$ is a spanning tree of G and $e \in E(\mathbf{x})$ if and only if $\mathbf{x}_e = 1$.

The data were generated at random following standard procedures as follows: let $s > 0$, $0 < dens < 1$ and $\gamma > 0$. A connected graph $G = (V,E)$ with $|V| = s$, density equal to $dens$ and random topology is generated. For each e, $\mathbf{L}_e$ is taken from $U(0,100)$ and $\mathbf{U}_e = (1+r_e\gamma)\mathbf{L}_e$ where r_e is taken from $U(0,1)$. The definitive data were obtained by rounding down the data generated.

We need a 0-1-Mixed Integer model to X. We use the known single commodity model as follows:

Let $V = \{1, \cdots, |V|\}$. If $e \in E$ then $e = \{i, j\}$ for some $i, j \in V$ with $i \neq j$. If $\{i, j\} \in E$ then we have two auxilary variables: $\mathbf{f}_{ij}$ and $\mathbf{f}_{ji}$. Let $F(X)$ a set in $(\mathbf{x}, \mathbf{f})$ defined as follows:

$$F(X) = \{(\mathbf{x}, \mathbf{f}) :$$

$$\sum_{j:\{1,j\}\in E} (\mathbf{f}_{1j} - \mathbf{f}_{j1}) = |V| - 1,$$

$$\sum_{j:\{i,j\}\in E} (\mathbf{f}_{ij} - \mathbf{f}_{ji}) = -1 \ \ \forall i \in V, \ i \neq 1,$$

$$\mathbf{f}_{ij} \leq (|V| - 1)\mathbf{x}_{\{i,j\}}, \ \ \mathbf{f}_{ji} \leq (|V| - 1)\mathbf{x}_{\{i,j\}} \ \ \forall \{i, j\} \in E,$$

$$\mathbf{x} \in \{0, 1\}^n, \ \ \mathbf{f}_{ij} \geq 0, \mathbf{f}_{ji} \geq 0 \ \ \forall \{i, j\} \in E, \ \ \mathbf{f} \in \mathbb{R}^{2n}\}$$

We know that $\mathbf{x} \in X$ if and only if there exists an unique $\mathbf{f}$ such that $(\mathbf{x}, \mathbf{f}) \in F(X)$.

Now we can rewrite the problems $P(\mathbf{c})$,$Q(H)$,$S(Y)$,$W(H, Y)$,$Q_r(H)$, $Q_r(\mu, H)$, $S_r(Y)$ and $W_r(H, Y)$ as a 0-1-Mixed Integer Linear Programming (0-1-MILP) problems in order to be solved by using CPLEX.

6.3 Performance of the algorithms

We use $\epsilon = 0.0001$ for all experiments. However the ϵ-optimal algorithms generally stop with $UB = LB$.

Results associated with the use of the S-Q and T-Greedy algorithms for MST problems with $k \in \{2, 3\}$ may be seen in table 1. The T-Greedy algorithm found optimal solutions for the eight problems when $k = 2$ with less computational effort, in general, than that associated with the S-Q algorithm. When $k = 3$ the T-Greedy algorithm found optimal solutions for five problems (1,2,5,6 and,7) and a near optimal solution for two problems (3 and 4). When k = 3 the computational effort of the T-Greedy algorithm was considerably lower than that of the S-Q algorithm. The S-Q algorithm failed to find an optimal solution for problem 8 with $k = 3$ because of a time limit (60000 seconds).

In table 2 we have sixteen MST problems and we evaluated again the performance of the S-Q and T-Greedy algorithms for $k = 2$. The S-Q algorithm failed to find an optimal solution for problem 12 and failed to find an optimal solution for problem 17 when the information obtained from the T-Greedy algorithm was not used. The T-Greedy algorithm found optimal solutions for fourteen problems and a near optimal solution for problem 24. The performance of the

Table 1: MST problems. $s = 50$. T-greedy and S-Q algorithms for the MMSR(k) problem

prob	dens	γ	t1	R(1)	k	tT(k)	t(k)	RT(k)	R(k)
1	0.15	0.15	20	59	2	9	162	31	31
					3	21	55527	30	30
2		0.20	7	113	2	3	10	59	59
					3	8	1913	55	55
3		0.25	10	121	2	4	83	77	77
					3	13	50284	70	66
4		0.30	5	119	2	3	8	69	69
					3	5	323	63	62
5	0.20	0.15	51	26	2	34	17	16	16
					3	49	74	13	13
6		0.20	4	30	2	9	9	20	20
					3	13	23	15	15
7		0.25	4	85	2	2	17	57	57
					3	4	105	46	46
8		0.30	27	139	2	7	325	80	80
					3	6	-	75	-

Table 2: MST problems. $s = 75$.T-Greedy and S-Q algorithms for the MMSR(k) problem

prob	dens	γ	t1	R(1)	tT(2)	t(2)	tU(2)	to(2)	RT(2)	R(2)
9	0.15	0.15	22	26	12	59	50	119	16	16
10		0.20	24	82	7	1281	1412	2279	59	59
11		0.25	51	34	47	402	1475	1475	19	19
12		0.30	1110	186	1100	-	-	-	123	≥ 95
13	0.20	0.15	7	20	8	15	21	48	12	12
14		0.20	24	40	15	62	194	227	24	24
15		0.25	64	31	60	936	2251	2830	20	20
16		0.30	39	83	17	194	70	436	45	45
17	0.25	0.15	112	27	71	23200	-	-	20	20
18		0.20	113	25	51	534	465	1581	13	13
19		0.25	184	22	153	1353	6449	6499	12	12
20		0.30	49	44	34	64	215	215	23	23
21	0.30	0.15	89	18	89	333	245	725	5	5
22		0.20	99	29	58	8638	248	13188	18	18
23		0.25	13	10	28	35	58	58	5	5
24		0.30	230	63	124	24681	409	23148	42	41

Table 3: MST problems. Q-Greedy and T-Greedy algorithms for the MMSR(k) problem

prob	s	dens	γ	t1	*	t*(5)	$red(*,2)$	$red(*,3)$	$red(*,4)$	$red(*,5)$
25	75	0.15	0.25	49	Q	18	8	13	17	20
					T	974	25	37	43	46
26			0.30	46	Q	3	6	7	8	16
					T	207	30	38	42	43
27		0.30	0.25	96	Q	7	7	12	12	12
					T	114	40	43	43	48
28			0.30	31	Q	3	19	23	27	29
.					T	95	27	38	44	48
29	80	0.15	0.25	267	Q	17	30	31	34	36
					T	418	35	42	43	46
30			0.30	191	Q	6	3	8	11	18
					T	591	15	20	27	31
31		0.30	0.25	47	Q	4	0	11	11	16
					T	92	42	53	53	63
32			0.30	2705	Q	60	10	18	26	33
					T	4480	35	39	43	44

Table 4: p-M problems. Real Data. T-Greedy and S-Q algorithms for the MMSR(k) problem

prob	n	p	a(b)	γ	t1	R(1)	tT(2)	t(2)	tU(2)	to(2)	RT(2)	R(2)
1	100	5	1(9)	0.25	5	1968.92	23	202	129	163	1792.09	1792.09
2				0.35	6	1460.44	10.3	179	35	155	977.86	977.86
3			2(8)	0.25	4	1880.72	18	125	7	193	1649.48	1642.46
4				0.35	14	5131.79	50	2281	409	1557	3777.68	3708.08
5		10	1(9)	0.25	7	310.89	8	14	60	73	101.90	101.90
6				0.35	9	734.40	5	151	79	103	441.65	441.65
7			2(8)	0.25	8	1707.10	10	318	320	356	1151.28	1151.28
8				0.35	51	1868.97	23	1835	2144	2284	1367.77	1367.77
9	125	5	1(9)	0.25	27	1104.56	91	965	14	169	797.40	685.48
10				0.35	8	2043.04	28	808	476	657	1278.47	1272.18
11			2(8)	0.25	29	3419.25	44	826	386	458	2410.18	2410.18
12				0.35	65	6670.65	165	-	-	-	6000.47	$\leq$ 5994.74 $\geq$ 5712.95
13		10	1(9)	0.25	16	492.46	17	161	83	136	283.37	283.37
14				0.35	22	1302.75	120	1955	1320	1527	1177.58	1177.58
15			2(8)	0.25	71	2046.03	504	37258	-	-	1604.04	1604.04
16				0.35	143	3563.52	645	-	-	-	3064.40	$\leq$ 3062.79 $\geq$ 2944.95

Table 5: p-M problems. Integer Data. Q-Greedy and T-Greedy algorithms for the MMSR(k) problem

prob	n	p	a(b)	γ	t1	tQ(5)	tT(5)
17	100	5	1(9)	0.25	5	7	28
18				0.35	20	5	52
19			2(8)	0.25	24	6	56
20				0.35	42	12	136
21	100	10	1(9)	0.25	20	4	39
22				0.35	13	5	41
23			2(8)	0.25	8	3	45
24				0.35	26	4	84
25	200	10	1(9)	0.25	89	94	659
26				0.35	311	45	925
27			2(8)	0.25	420	52	1834
28				0.35	1581	94	544
29	200	20	1(9)	0.25	213	39	1782
30				0.35	427	34	1347
31			2(8)	0.25	1232	56	18635
32				0.35	1616	45	17690

S-Q algorithm result much better when the information obtained by using the T-Greedy algorithm was used. Again, the computational effort of the T-Greedy algorithm result considerably lower than that of the S-Q algorithm.

In table 3 we have eight MST problems and we evaluated the performance of the T-Greedy and Q-Greedy algorithms for $k \in \{2, 3, 4, 5\}$. In the table we can see the percentages of reduction of the regret values associated to the two algorithms. As expected the reduction percentages and the computational effort are larger using the T-Greedy algorithm. A good news is that the percentage of reduction are large in general.

In table 4 we present sixteen p-M problems and the performance of the T-Greedy and the S-Q algorithms is considered again when $k = 2$. The S-Q algorithm failed to find an optimal solution for problems 12 and 16 and failed to find an optimal solution for problem 15 when the information obtained from the T-Greedy algorithm was not used. The T-Greedy algorithm found optimal solutions for ten problems and a near optimal solution for problems 3,4,10,12,and 16. The T-Greedy algorithm failed to find a good solution for problem 9. Again, the computational effort of the T-Greedy algorithm result considerably lower than that of the S-Q algorithm. For these problems the use of the information previously generated for the T-greedy algorithm to use the S-Q algorithm proved unnecessary.

Table 6: p-M problems. Q-Greedy and T-Greedy algorithms for the MMSR(k) problem. Integer Data.

prob	*	$red(*,2)$	$red(*,3)$	$red(*,4)$	$red(*,5)$
17	Q	38	45	46	51
	T	38	51	55	58
18	Q	12	19	29	29
	T	17	32	41	42
19	Q	24	42	43	46
	T	34	43	45	46
20	Q	7	8	20	21
	T	7	13	21	23
21	Q	6	13	16	21
	T	23	34	35	37
22	Q	8	22	24	26
	T	15	26	31	40
23	Q	12	38	45	45
	T	32	41	46	48
24	Q	10	22	23	28
	T	20	30	33	34
25	Q	9	20	20	23
	T	9	20	21	31
26	Q	4	7	15	16
	T	13	17	21	23
27	Q	1	8	9	14
	T	5	11	14	17
28	Q	7	12	16	17
	T	17	20	21	22
29	Q	7	13	16	17
	T	16	20	22	25
30	Q	8	15	15	19
	T	15	23	25	28
31	Q	8	17	21	23
	T	10	21	27	29
32	Q	10	16	18	18
	T	13	17	19	21

Table 7: SC problems. Tr-Greedy algorithm for the MMrelSR(k) problem

prob	n	K	γ	t1r	tTr(5)	$red(T_r, 2)$	$red(T_r, 3)$	$red(T_r, 4)$	$red(T_r, 5)$
1	100	5	0.15	1	5	25	34	37	38
2			0.30	1	12	30	36	39	40
3		10	0.15	1	1	28	59	62	65
4			0.30	3	3	15	21	22	25
5	200	5	0.15	3	6	33	44	47	48
6			0.30	105	117	18	22	25	27
7		10	0.15	3	7	9	9	21	26
8			0.30	3	11	14	17	19	22
9		20	0.15	2	6	3	11	16	19
10			0.30	2	5	3	7	8	20
11	300	10	0.15	24	73	7	14	17	21
12			0.30	45	90	6	9	11	12
13		15	0.15	7	17	31	39	41	47
14			0.30	29	275	14	18	21	23
15		30	0.15	3	17	20	28	38	42
16			0.30	2	10	8	17	22	24
17	600	20	0.15	50	4335	13	13	15	16
18			0.30	3775	52629	5	8	10	11

Table 8: p-M problems. Tr-Greedy and Qr-Greedy algorithms for the MMrelSR(k) problem

prob	n	p	(a,b)	γ	t1r	tTr(5)	tQr(5)
33	100	5	(1,9)	0.25	31	54	6
34				0.35	28	52	6
35			(2,8)	0.25	36	54	8
36				0.35	30	141	15
37		10	(1,9)	0.25	11	43	5
38				0.35	11	33	6
39			(2,8)	0.25	18	71	9
40				0.35	21	107	12
41	200	10	(1,9)	0.25	112	866	94
42				0.35	184	1499	100
43			(2,8)	0.25	565	6925	225
44				0.35	1161	5669	777
45		20	(1,9)	0.25	273	863	66
46				0.35	2326	11472	92
47			(2,8)	0.25	475	3751	54
48				0.35	8156	23480	102

Table 9: p-M problems. Tr-Greedy and Qr-Greedy algorithms for the MM-relSR(k) problem

prob	*	$red(*,2)$	$red(*,3)$	$red(*,4)$	$red(*,5)$
33	Tr	38	45	53	54
	Qr	20	29	45	46
34	Tr	40	46	47	48
	Qr	37	39	40	43
35	Tr	23	28	32	37
	Qr	17	18	26	30
36	Tr	17	32	33	34
	Qr	5	13	25	27
37	Tr	35	54	64	64
	Qr	10	17	27	35
38	Tr	28	38	42	49
	Qr	14	27	28	34
39	Tr	24	34	38	39
	Qr	5	26	33	37
40	Tr	12	15	16	22
	Qr	1	14	15	15
41	Tr	17	23	28	33
	Qr	17	21	23	28
42	Tr	8	11	12	13
	Qr	0	3	4	5
43	Tr	8	13	15	20
	Qr	2	5	13	16
44	Tr	10	11	13	15
	Qr	10	11	12	13
45	Tr	22	30	35	39
	Qr	14	24	30	30
46	Tr	14	21	25	28
	Qr	9	12	19	21
47	Tr	15	22	24	25
	Qr	3	5	9	11
48	Tr	14	18	20	21
	Qr	3	7	10	12

Table 10: SP problems. Tr-Greedy algorithm for the MMrelSR(k) problem

prob	n	m	γ	t1r	tTr(5)	$red(T_r, 2)$	$red(T_r, 3)$	$red(T_r, 4)$	$red(T_r, 5)$
1	10	25	0.15	1	1	100	100	100	100
2			0.35	1	3	54	75	85	89
3		50	0.15	1	5	47	53	55	55
4			0.35	3	18	16	26	29	30
5		75	0.15	1	1	100	100	100	100
6			0.35	8	18	38	42	46	49
7	15	25	0.15	1	3	29	37	47	58
8			0.35	1	10	56	54	100	100
9		50	0.15	1	1	0	0	0	0
10			0.35	1	10	20	32	32	42
11		75	0.15	5	30	20	31	36	40
12			0.35	9	42	23	32	37	40
13	30	25	0.15	2	11	24	26	64	75
14			0.35	6	25	27	38	42	45
15		50	0.15	9	48	45	66	67	69
16			0.35	14	68	34	43	44	45
17		75	0.15	33	106	27	31	37	41
18			0.35	75	435	22	25	32	35
19	60	25	0.15	10	92	47	90	100	100
20			0.35	21	110	5	23	32	34
21		50	0.15	24	264	18	39	52	60
22			0.35	65	242	14	19	19	21
23		75	0.15	116	520	41	53	55	60
24			0.35	239	1120	25	30	34	37

Tables 5 and 6 refer to the use of T-Greedy and Q-Greedy algorithms for sixteen p-M problems when $k \in \{2, 3, 4, 5\}$. Table 5 refers to the computational effort and table 6 refers to the percentages of reduction of the regret values. In table 5 may be seen that the computational effort increases as dimensions and uncertainty increase. Again the reduction percentages and the computational effort are larger using the T-Greedy algorithm and the percentage of reduction are large in general.

In table 7 we present results about the Tr-Greedy algorithm for eighteen SC problems with $k \in \{2, 3, 4, 5\}$. The percentage of reduction of the relative regret values are large in general.

Tables 8 and 9 refer to the use of Tr-Greedy and Qr-Greedy algorithms for sixteen p-M problems when $k \in \{2, 3, 4, 5\}$. Table 8 refers to the computational effort and table 9 refers to the percentages of reduction of the relative regret values. In table 8 may be seen that the computational effort increases as dimensions and uncertainty increase. As expected the reduction percentages and the computational effort are larger using the Tr-algorithm and again the percentage of reduction are large in general.

In table 10 we present results about the Tr-Greedy algorithm for twenty four SP problems with $k \in \{2, 3, 4, 5\}$. The percentage of reduction of the relative regret values are large in general.

As we can expect the computational effort increases, in general, while the dimensions and uncertainty increase. However more problems with different structures and without known structures must be solved.

We present results for 122 problems. The greedy approaches worked well for all problems: the sets generated had a considerable percentage reduction of the (relative) regret values with respect to the single (relative) absolute robust solution.

Note that the time to find the first solution (either an absolute or a relative robust solution) and the time to find the remaining solutions are presented separately. This is because for some problems, the SP problems (Montemanni and Gambardella 2005(A)) and the MST problems (Montemanni and Gambardella 2005(B)) in our experiments, the algorithms for $k = 1$ may be changed to specialized algorithms. That is not the case, at the present time, for the algorithms with $k > 1$.

7 Conclusions and further extensions

7.1 Conclusions

We presented the regret and the relative regret of a set with k elements. Algorithms to find a set with the minimun (relative) regret were developed. Also, greedy approaches to save computational effort were designed. Our approach is a generalization of the min max min approach recently presented (Buchheim and Kurtz 2016, Buchheim and Kurtz J 2017).

S-Q and Sr-Qr algorithms defined to solve the MMSR(k) and the MMrelSR(k) problem respectively may be seen as a generalization of classical algorithms to obtain an absolute and relative robust solution.

Our greedy approach include two algorithms , the first one based on conditionated absolute (relative) robust solutions (T-Greedy) (Tr-Greedy) and the second one based on a simple mutiparametric algorithm (Q-Greedy) (Qr-Greedy).

We presented a computational experience for the p-M,MST,SC and SP problems in order to point out that the approach may be computationally viable (offline) and now must be clear that our algorithms may be used, for small values of k, to obtain a set of solutions, instead of a single one, with better performance.

Based on our computational experiences we can expect that:

1. for $k \leq 3$ the S-Q algorithmw may be used offline with tolerables execution times,
2. for $k \leq 3$ the T-Greedy algorithm obtains optimal or near optiomal solutions, and for $k \leq 5$ the T-Greedy and the Tr-Greedy algorithm obtains a good set of solutions with a tolerable computational effort,
3. the Q-Greedy and Qr-Greedy algorithms may be used to obtain a good set of solutions quickly.

Finally, the tradeoff between the computational effort and the quality of the solutions generated with our algorithms must be considered by the decision maker.

7.2 Extensions

In robust programming a significant effort is directed towards the design of special purpose algorithms in order to find robust solutions to problems with particular structures. It is reasonable then to think that a next step should be the design of specialized algorithms to solve the MMSR(k) and MMrelSR(k) problems.

The approach used to consider the Minimun Spanning Tree problem may be generalized to any 0-1-MILP problem. The paper may be rewritten without any problem if $P(\mathbf{c})$ is a 0-1-MILP problem with the uncertainty relative to the cost of the 0-1-variables as follows:

$$(P(\mathbf{c}))\ \ \min\{\mathbf{c}^t\mathbf{x}+\mathbf{d}^t\mathbf{y} : (\mathbf{x},\mathbf{y}) \in X, \mathbf{x} \in \{0,1\}^n, \mathbf{y} \in \mathbb{R}^m, \mathbf{y} \geq 0\}$$

where $\mathbf{c} \in [\mathbf{L}, \mathbf{U}]$, $\mathbf{d} \in \mathbb{R}^m$ and $X \subseteq \{0,1\}^n \times \mathbb{R}^m$.

The algorithms may be improved with a better choice for the Big-M values. A dynamic choice may be designed by using standard procedures (Crema A (2014): Mathematical programming approach to tighten a Big-M formulation, *www.optimization-online.org*).

We must pointed out that our approach may be used in practice if the cost vectors become known over the time. That may be the situation if the travel times and the demands of a emergency system are monitored in real time. In that scenario we can choose a new solution when the system changes enough. Multiples changes in a short time, even by using a small set of solutions, may be a problem for humans and that should be consider for further works.

References

[1] Aissi H, Bazgan C, Vanderpooten D (2009) Min?max and min?max regret versions of combinatorial optimization problems: A survey. *Eur. J. Oper. Res.* 197(2)427-438.

[2] Averbakh I (2005) Computing and minimizing the relative regret in combinatorial optimization with interval data *Discrete Optim.*2:273-287

[3] Boffey TB, Karkazis J (1984) p-Medians and Multi-Medians *J. Oper. Res. Soc.* 35(1)57-64.

[4] Buchheim C, Kurtz J (2016) Min-max-min robustness: a new approach to combinatorial optimization under uncertainty based on multiple solutions *Electronic Notes in Discrete Mathematics* 52:45-52.

[5] Buchheim C, Kurtz J (2017) Robust combinatorial optimization *Math. Program.* 163(1?2)1-23.

[6] Candia-Véjar A, Alvarez-Miranda E, Maculan N (2011) Minmax regret combinatorial optimization problems: an alogorithmic perspective *RAIRO-Oper. Res.* 45:101?129.

[7] Caprara A, Toth P, Fischetti M (2000) Algorithms for the Set Covering Problem *Ann. Oper. Res.* 98:353-371.

[8] Crema A (2000) An algorithm for the multiparametric 0-1 integer linear programming *Eur. J. of Oper Res.* 125:18-24.

[9] Kasperski A, Zielinski P (2016) Robust Discrete Optimization Under Discrete and Interval Uncertainty: A Survey. A Chapter in Robustness Analysis in Decision Aiding, Optimization, and Analytics. Editors: Doumpos M, Zopounidis C, Grigoroudis E. International. Series in Operations Research & Management Science, Springer.

[10] Li J, Liu Y (2016) Approximation Algorithms for Stochastic Combinatorial Optimization Problems *J. Oper. Res. Soc. China* 4(1)1-47.

[11] Megiddo N (1979) Combinatorial optimization with fractional objective functions *Math. Oper. Res* 4(4)414-424.

[12] Montemanni R, Gambardella LM (2005(A)) A branch and bound algorithm for the robust spanning tree problem with interval data *Eur. J. Oper. Res* 161(3)771-779.

[13] Montemanni R, Gambardella LM (2005(B)) The robust shortest path problem with interval data via Benders decomposition *4OR* 3:315-328.

[14] Oberdieck R, Diangelakis NA, Nascu I, Papathanasiou MM, Sun M, Avraamidou S, and Pistikopoulos EN (2016) On multi-parametric programming and its applications in process systems engineering *Chem. Eng. Res. Des.* 116:61-82.