



Min max min robust (relative) regret combinatorial optimization

Alejandro Crema¹

Received: 20 March 2019 / Revised: 25 March 2020 / Published online: 19 May 2020
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract

We consider combinatorial optimization problems with uncertainty in the cost vector. Recently, a novel approach was developed to deal with 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 apply the new approach to the absolute and relative regret cases. Algorithms to solve the min max min robust (relative) regret problems are presented with computational experiments.

Keywords Minmax regret · Robust programming · Greedy algorithms · Multiparametric programming

1 Introduction

Data uncertainty appears in many optimization problems. In recent decades *Robust* (Aissi et al. 2009; Averbakh 2005; Candia-Véjar et al. 2011; Kouvelis and Yu 2013; Chassein and Goerigk 2016; Kasperski and Zielinski 2016), *Stochastic* (Li and Liu 2016), *Multiparametric* (Oberdieck et al. 2016) and *fuzzy programming* (Carlsson and Fuller 2012) approaches have been developed to deal with such uncertainties. In this paper we use the Robust approach.

There are several robust optimization concepts you may select. Some examples are (Chassein and Goerigk 2016): classic robustness, absolute or relative regret robustness, adjustable robustness, recoverable robustness, light robustness, soft robustness,

✉ Alejandro Crema
alejandro.crema@ciens.ucv.ve

¹ Escuela de Computación, Facultad de Ciencias, Universidad Central de Venezuela, Caracas, Venezuela

lexicographic α -robustness, recovery-to-optimality, or similarity-based robustness. In this paper we consider the classic robustness and absolute or relative regret.

Let P be a combinatorial optimization problems in (\mathbf{x}) with uncertainty in the cost vector as follows:

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

where $\mathbf{c} \in \Omega$, $\Omega \subseteq \mathbb{R}^n$, $X \subseteq \{0, 1\}^n$, X is not an empty set.

A few words about our notation: If R is an optimization problem then $v(R)$ is its optimal value and let $[n] = \{1, \dots, n\}$ denote a set of indices.

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

$$\min_{\mathbf{x} \in X} \max_{\mathbf{c} \in \Omega} \mathbf{c}^t \mathbf{x}.$$

Recently a novel approach has been presented considering a set of k solutions instead of a single one (Buchheim and Kurtz 2016, 2017; Chassein et al. 2019). The problem in (H) to be solved is:

$$\min_{H \subseteq X} \left\{ \max_{\mathbf{c} \in \Omega} \min_{\mathbf{h} \in H} \mathbf{c}^t \mathbf{h} : |H| = k \right\}.$$

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.

The regret approach, instead of classical robust approach, has been considered by many authors (Chassein and Goerigk 2016) as a valid option for the $k = 1$ case in order to take into account the deviation from the optimal values.

An approach of robust optimization taking into account the absolute deviation from the optimal values is to find a solution with a minimum *regret* known as an *absolute robust solution*. The regret of \mathbf{x} is equal to

$$\max_{\mathbf{c} \in \Omega} \mathbf{c}^t \mathbf{x} - v(P(\mathbf{c}))$$

In this case the problem to be solved is the *min max regret* problem in (\mathbf{x}) :

$$\min_{\mathbf{x} \in X} \max_{\mathbf{c} \in \Omega} \mathbf{c}^t \mathbf{x} - v(P(\mathbf{c}))$$

An approach of robust optimization taking into account the relative deviation from the optimal values is to find a solution with a minimum *relative regret* known as a *relative robust solution*. The relative regret of \mathbf{x} is equal to

$$\max_{\mathbf{c} \in \Omega} \frac{\mathbf{c}^T \mathbf{x} - v(P(\mathbf{c}))}{v(P(\mathbf{c}))}$$

In this case the problem to be solved is the *min max relative regret* problem in (\mathbf{x}) :

$$\min_{\mathbf{x} \in X} \max_{\mathbf{c} \in \Omega} \frac{\mathbf{c}^T \mathbf{x} - v(P(\mathbf{c}))}{v(P(\mathbf{c}))}$$

To the best of our knowledge the regret approach with $k > 1$ solutions was not studied so far. Our main contribution is to close this gap . In this paper, we extend the novel approach presented above considering a set of k solutions and take into account the absolute and relative deviations from the optimal values.

Let $H \subseteq X$ with $|H| \geq 1$. Let $Q(H)$ be a problem in (\mathbf{c}) defined as follows:

$$(Q(H)) \max_{\mathbf{c} \in \Omega} \min_{\mathbf{h} \in H} \mathbf{c}^T \mathbf{h} - v(P(\mathbf{c}))$$

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

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

$$(Q_r(H)) \max_{\mathbf{c} \in \Omega} \frac{\min_{\mathbf{h} \in H} \mathbf{c}^T \mathbf{h} - v(P(\mathbf{c}))}{v(P(\mathbf{c}))}$$

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 minimum regret (minimum relative regret) by solving the min max min robust (relative) regret problems, namely: *min set-regret(k)* problem (*min relative set-regret(k)* problem) in (H) , defined as follows:

$$\begin{aligned} (\text{MSR}(k)) \quad & \min_{H \subseteq X} \{v(Q(H)) : |H| = k\} \\ (\text{MrelSR}(k)) \quad & \min_{H \subseteq X} \{v(Q_r(H)) : |H| = k\} \end{aligned}$$

Note that if we use $1 \leq |H| \leq k$ instead of $|H| = k$ we have exactly the same problems.

There are several options to model the uncertainty. Some examples are (Chassein and Goerigk 2016): finite uncertainty, interval uncertainty, bounded uncertainty, ellipsoidal uncertainty and polytopal uncertainty.

Let us consider the interval uncertainty case as follows: Let $\Omega = \{\mathbf{c} : \mathbf{L} \leq \mathbf{c} \leq \mathbf{U}\}$, $\mathbf{L} \in \mathbb{R}^n, \mathbf{U} \in \mathbb{R}^n$ with $\mathbf{0} \leq \mathbf{L} \leq \mathbf{U}$. Let us suppose that $\mathbf{c}^T \mathbf{x} > 0$ for all $\mathbf{c} \in \Omega$ and for all $\mathbf{x} \in X$. Note that $v(P(\mathbf{c})) > 0$ for all $\mathbf{c} \in \Omega$.

For the interval uncertainty case the min max approach (for $k = 1$) and the min max min approach (for $k > 1$) can be reduced easily to solve $P(\mathbf{U})$.

In Sect. 2, we present an ϵ -optimal algorithm (with $\epsilon \geq 0$ a predefined tolerance) to solve the MSR(k) problem that may be considered a generalization of a classical approach where $k = 1$ (Aissi et al. 2009). Solving the MSR(k) problem it could be a task that consumes a lot of computing time and a lot of memory, therefore it may be useful to consider two greedy approaches, which will be presented in Sect. 3.

In Sect. 4, we consider the MreISR(k) problem and we present an ϵ -optimal algorithm to solve it. Our presentation follows the same scheme that we used for the MSR(k) problem and the proof of some lemmas are analogous. Thus, for the sake of simplicity and in order to save space, we omit some proofs. Again, two greedy approaches will be presented in Sect. 5.

Computational results are presented in Sect. 6 for the Shortest Path problem (Montemanni and Gambardella 2005) and the p -Medians problem. Conclusions and further extensions are presented in Sect. 7.

2 An algorithm to solve the MSR(k) problem

In this section, we present an algorithm to obtain an optimal solution for the MSR(k) problem. If $H \subseteq X$ with $|H| = k$ then $v(Q(H))$ is an upper bound of $v(\text{MSR}(k))$. Let $Y \subseteq X$. We present later a problem $S(Y)$ in (H) to obtain a lower bound. The algorithm iterates between solving the problems $S(Y)$ (to generate a new H) and $Q(H)$ (to generate a new \mathbf{y} to be added to Y) until the gap is closed. If $k = 1$ the algorithm becomes a known classical algorithm to obtain an absolute robust solution. Some lemmas and auxiliary problems are necessary.

Let $\mathbf{x} \in X$. The *best scenario* for \mathbf{x} is defined as follows:

$$\mathbf{c}^+(\mathbf{x})_j = \mathbf{L}_j \mathbf{x}_j + \mathbf{U}_j(1 - \mathbf{x}_j) \text{ for all } j$$

We need some well known elementary properties as follows (for the sake of completeness we present a proof):

- Proposition 1** (i) $\mathbf{c}'\mathbf{h} - \mathbf{c}'\mathbf{x} \leq \mathbf{c}^+(\mathbf{x})'\mathbf{h} - \mathbf{c}^+(\mathbf{x})'\mathbf{x} \quad \forall \mathbf{c} \in \Omega \quad \forall \mathbf{x}, \mathbf{h} \in X$
 (ii) If $\mathbf{c}^+(\mathbf{x})'\mathbf{x} > v(P(\mathbf{c}^+(\mathbf{x})))$ then $\mathbf{c}'\mathbf{x} > v(P(\mathbf{c})) \quad \forall \mathbf{x} \in X \quad \forall \mathbf{c} \in \Omega$
 (iii) $\mathbf{c}^+(\mathbf{x})'\mathbf{x} = \mathbf{L}'\mathbf{x} \leq \mathbf{c}'\mathbf{x} \quad \forall \mathbf{c} \in \Omega \quad \forall \mathbf{x} \in X$

Proof See ‘‘Appendix A’’.

As an immediate consequence of Proposition 1(ii) we have that \mathbf{x} is an optimal solution for $P(\mathbf{c})$ for some $\mathbf{c} \in \Omega$ if and only if \mathbf{x} is an optimal solution for $P(\mathbf{c}^+(\mathbf{x}))$. That suggests to solve $Q(H)$ the search can be restricted to scenarios of the type $\mathbf{c}^+(\mathbf{x})$ such that \mathbf{x} is an optimal solution for $P(\mathbf{c}^+(\mathbf{x}))$. That is proved in Lemma 1(i), (ii). In order to solve $Q(H)$ we present in Lemma 1(iii) a reformulation named $Q2(H)$ by using a level set transformation. The relation between the optimal solutions for both problems is presented in Lemma 2. □

Lemma 1 Let $H \subseteq X$ with $|H| \geq 1$, then:

- (i) $v(Q(H)) = \max_{\mathbf{x} \in X} \min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{x})^t \mathbf{h} - \mathbf{c}^+(\mathbf{x})^t \mathbf{x}$
- (ii) Let $\mathbf{x}^* \in X$. If $v(Q(H)) = \min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{x}^*)^t \mathbf{h} - \mathbf{c}^+(\mathbf{x}^*)^t \mathbf{x}^*$ then \mathbf{x}^* is an optimal solution for $P(\mathbf{c}^+(\mathbf{x}^*))$.
- (iii) Given the problem $Q2(H)$ in (σ, \mathbf{x}) defined as follows:

$$(Q2(H)) \max_{\sigma \in \mathbb{R}, \mathbf{x} \in X} \{ \sigma - \mathbf{L}^t \mathbf{x} : \sigma \leq \mathbf{c}^+(\mathbf{x})^t \mathbf{h} \quad \forall \mathbf{h} \in H \}$$

then $v(Q(H)) = v(Q2(H))$.

Proof

$$\begin{aligned} \text{(i): } v(Q(H)) &\stackrel{(1)}{=} \max_{\mathbf{c} \in \Omega} \min_{\mathbf{h} \in H} \mathbf{c}^t \mathbf{h} - v(P(\mathbf{c})) \stackrel{(2)}{=} \\ &\max_{\mathbf{c} \in \Omega, \mathbf{x} \in X} \min_{\mathbf{h} \in H} \mathbf{c}^t \mathbf{h} - \mathbf{c}^t \mathbf{x} \stackrel{(3)}{\leq} \\ &\max_{\mathbf{x} \in X} \min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{x})^t \mathbf{h} - \mathbf{c}^+(\mathbf{x})^t \mathbf{x} \stackrel{(4)}{\leq} v(Q(H)) \end{aligned}$$

where

- (1) From the definition of $Q(H)$.
- (2) From the definition of $P(c)$.
- (3) From Proposition 1(i).
- (4) From the definition of $Q(H)$ and since $\mathbf{c}^+(\mathbf{x}) \in \Omega \quad \forall \mathbf{x} \in X$.
- (ii) Let us suppose that $\hat{\mathbf{x}} \in X$ is an optimal solution for $P(\mathbf{c}^+(\mathbf{x}^*))$ with $\mathbf{c}^+(\mathbf{x}^*)^t \hat{\mathbf{x}} < \mathbf{c}^+(\mathbf{x}^*)^t \mathbf{x}^*$. In that case we have that:
 $v(Q(H)) = \min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{x}^*)^t \mathbf{h} - \mathbf{c}^+(\mathbf{x}^*)^t \mathbf{x}^* < \min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{x}^*)^t \mathbf{h} - \mathbf{c}^+(\mathbf{x}^*)^t \hat{\mathbf{x}} =$
 $\min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{x}^*)^t \mathbf{h} - v(P(\mathbf{c}^+(\mathbf{x}^*))) \leq v(Q(H))$ and we have a contradiction.
- (iii) Since $\mathbf{c}^+(\mathbf{x})^t \mathbf{x} = \mathbf{L}^t \mathbf{x}$ for all $\mathbf{x} \in X$ we have that

$$\begin{aligned} v(Q(H)) &= \max_{\mathbf{x} \in X} \min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{x})^t \mathbf{h} - \mathbf{c}^+(\mathbf{x})^t \mathbf{x} \\ &= \max_{\sigma \in \mathbb{R}, \mathbf{x} \in X} \{ \sigma - \mathbf{L}^t \mathbf{x} : \sigma \leq \mathbf{c}^+(\mathbf{x})^t \mathbf{h} \quad \forall \mathbf{h} \in H \} = v(Q2(H)) \bullet \end{aligned}$$

□

Lemma 2 Let $H \subseteq X$ with $|H| \geq 1$.

- (i) Let \mathbf{c}^* be an optimal solution for $Q(H)$, let \mathbf{x}^* be an optimal solution for $P(\mathbf{c}^*)$ and let $\sigma^* = \min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{x}^*)^t \mathbf{h}$ then (σ^*, \mathbf{x}^*) is an optimal solution for $Q2(H)$.
- (ii) Let (σ^*, \mathbf{x}^*) be an optimal solution for $Q2(H)$ then $\mathbf{c}^+(\mathbf{x}^*)$ is an optimal solution for $Q(H)$.

Proof

$$\begin{aligned}
 (i) \quad v(Q(H)) &\stackrel{(1)}{=} \min_{\mathbf{h} \in H} \mathbf{c}^{*T} \mathbf{h} - v(P(\mathbf{c}^*)) \stackrel{(2)}{=} \min_{\mathbf{h} \in H} \mathbf{c}^{*T} \mathbf{h} - \mathbf{c}^{*T} \mathbf{x}^* \stackrel{(3)}{\leq} \\
 &\min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{x}^*)^T \mathbf{h} - \mathbf{c}^+(\mathbf{x}^*)^T \mathbf{x}^* \stackrel{(4)}{=} \sigma^* - \mathbf{L}^T \mathbf{x}^* \stackrel{(5)}{\leq} v(Q2(H)) \stackrel{(6)}{=} v(Q(H))
 \end{aligned}$$

where

- (1) Since \mathbf{c}^* is an optimal solution for $Q(H)$.
- (2) Since \mathbf{x}^* is an optimal solution for $P(\mathbf{c}^*)$.
- (3) Since $\mathbf{c}^T \mathbf{h} - \mathbf{c}^T \mathbf{x} \leq \mathbf{c}^+(\mathbf{x})^T \mathbf{h} - \mathbf{c}^+(\mathbf{x})^T \mathbf{x}$ for all $\mathbf{h} \in X, \mathbf{x} \in X, \mathbf{c} \in \Omega$.
- (4) From the definition of σ^* and since $\mathbf{c}^+(\mathbf{x})^T \mathbf{x} = \mathbf{L}^T \mathbf{x} \quad \forall \mathbf{x} \in X$.
- (5) Since (σ^*, \mathbf{x}^*) is a feasible solution for $Q2(H)$.
- (6) From Lemma 1.

Therefore (σ^*, \mathbf{x}^*) is an optimal solution for $Q2(H)$.

$$(ii) \quad v(Q(H)) \stackrel{(1)}{=} v(Q2(H)) \stackrel{(2)}{=} \sigma^* - \mathbf{L}^T \mathbf{x}^* \stackrel{(3)}{=} \min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{x}^*)^T \mathbf{h} - \mathbf{c}^+(\mathbf{x}^*)^T \mathbf{x}^*$$

where

- (1) From Lemma 1.
- (2) Since (σ^*, \mathbf{x}^*) is an optimal solution for $Q2(H)$.
- (3) From the definition of $Q2(H)$ and since $\mathbf{L}^T \mathbf{x} = \mathbf{c}^+(\mathbf{x})^T \mathbf{x} \quad \forall \mathbf{x} \in X$.

Therefore from Lemma 1 the scenario $\mathbf{c}^+(\mathbf{x}^*)$ is an optimal solution for $Q(H)$ •.

We have designed $Q2(H)$ to compute upper bounds to $v(MSR(k))$. If we have another problem to compute lower bounds then we may design an algorithm that iterates between both problems until the gap is closed. To do that we reformulate $MSR(k)$ by using a level set transformation. In the reformulation we have one constraint for each $\mathbf{x} \in X$. Let $Y \subseteq X$. If $\mathbf{x} \notin Y$ we delete the constraint and we have a relaxation named $S(Y)$. Problem $S(Y)$ is presented in Lemma 3. □

Lemma 3 *Let $Y \subseteq X$ with $|Y| \geq 1$. Let $S(Y)$ be a problem in (δ, H) defined as follows:*

$$(S(Y)) \quad \min_{\delta \in \mathbb{R}, H \subseteq X} \{ \delta : \delta \geq \mathbf{c}^+(\mathbf{x})^T \mathbf{h} - \mathbf{L}^T \mathbf{x} \text{ for some } \mathbf{h} \in H \quad \forall \mathbf{x} \in Y, \quad |H| = k \}$$

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

Proof

$$\begin{aligned}
 v(\text{MSR}(k)) &\stackrel{(1)}{=} \min_{H \subseteq X} \{v(Q(H)) : |H| = k\} \stackrel{(2)}{=} \\
 &\min\{\max_{\mathbf{x} \in X} \min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{x})^t \mathbf{h} - \mathbf{L}^t \mathbf{x} : H \subseteq X, |H| = k\} \stackrel{(3)}{=} \\
 &\min_{\sigma \in \mathbb{R}, H \subseteq X} \{\delta : \delta \geq \min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{x})^t \mathbf{h} - \mathbf{L}^t \mathbf{x} \quad \forall \mathbf{x} \in X, |H| = k\} \stackrel{(4)}{\geq} \\
 &\min_{\sigma \in \mathbb{R}, H \subseteq X} \{\delta : \delta \geq \min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{x})^t \mathbf{h} - \mathbf{L}^t \mathbf{x} \quad \forall \mathbf{x} \in Y, |H| = k\} \stackrel{(5)}{=} \\
 &\min_{\delta \in \mathbb{R}, H \subseteq X} \{\delta : \delta \geq \mathbf{c}^+(\mathbf{x})^t \mathbf{h} - \mathbf{L}^t \mathbf{x} \text{ for some } \mathbf{h} \in H \quad \forall \mathbf{x} \in Y, |H| = k\} = v(S(Y))
 \end{aligned}$$

where

- (1) From the definition of MSR(k).
- (2) From Lemma 1.
- (3) By level set transformation.
- (4) Since $Y \subseteq X$.
- (5) A trivial equality.

Problem $S(Y)$ has some disjunctive constraints and we need a reformulation to solve it. In order to solve $S(Y)$ we use the *Big-M* reformulation named $S2(Y)$. $S(Y)$ may be rewritten as a problem in $(\delta, H, \mathbf{s}, \mathbf{z})$ as follows:

$$\begin{aligned}
 (S2(Y)) \quad &\min \delta \quad s.t. \\
 &\delta \geq \mathbf{s}_{\mathbf{x}} - \mathbf{L}^t \mathbf{x} \quad \forall \mathbf{x} \in Y \\
 &\mathbf{s}_{\mathbf{x}} \geq \mathbf{c}^+(\mathbf{x})^t \mathbf{h}^j - (1 - \mathbf{z}_{(j,\mathbf{x})})M_{\mathbf{x}} \quad \forall \mathbf{x} \in Y \quad \forall j \in [k] \\
 &\sum_{j=1}^k \mathbf{z}_{(j,\mathbf{x})} = 1 \quad \forall \mathbf{x} \in Y \\
 &\mathbf{h}^{(j)} \in X \quad \forall j \in [k], \quad \mathbf{z}_{(j,\mathbf{x})} \in \{0, 1\} \quad \forall \mathbf{x} \in Y \quad \forall j \in [k], \quad \delta \in \mathbb{R}, \quad \mathbf{s}_{\mathbf{x}} \in \mathbb{R} \quad \forall \mathbf{x} \in Y
 \end{aligned}$$

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

Let $\mathbf{x} \in Y$. Note that if $\mathbf{z}_{j,\mathbf{x}} = 1$ then $\mathbf{s}_{\mathbf{x}} \geq \mathbf{c}^+(\mathbf{x})^t \mathbf{h}^j$ and we have $\delta \geq \mathbf{c}^+(\mathbf{x})^t \mathbf{h} - \mathbf{L}^t \mathbf{x}$ for some $\mathbf{h} \in H$. If $\mathbf{z}_{j,\mathbf{x}} = 0$ then the big-M value ensures that the constraint is satisfied. The relations between the optimal solutions for $S(Y)$ and $S2(Y)$ are presented in Lemma 4. □

Lemma 4 (i) If $(\delta, H, \mathbf{s}, \mathbf{z})$ is a feasible solution for $S2(Y)$ then (δ, H) is a feasible solution for $S(Y)$. (ii) If (δ, H) is a feasible solution for $S(Y)$ then there exists (\mathbf{s}, \mathbf{z}) such that $(\delta, H, \mathbf{s}, \mathbf{z})$ is a feasible solution for $S2(Y)$. (iii) $v(S(Y)) = v(S2(Y))$

Proof (i) Let $(\delta, H, \mathbf{s}, \mathbf{z})$ be a feasible solution for $S2(Y)$ then if $\mathbf{z}_{(j,\mathbf{x})} = 1$ we have $\mathbf{s}_{\mathbf{x}} \geq \mathbf{c}^+(\mathbf{x})^t \mathbf{h}^j$ and then $\delta \geq \mathbf{c}^+(\mathbf{x})^t \mathbf{h} - \mathbf{L}^t \mathbf{x}$ for some $\mathbf{h} \in H \quad \forall \mathbf{x} \in Y$. Therefore (δ, H) is a feasible solution for $S(Y)$.

- (ii) Let (δ, H) be a feasible solution for $S(Y)$. Let $j(\mathbf{x})$ and \mathbf{s}_x such that $\mathbf{s}_x = \min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{x})^t \mathbf{h} = \mathbf{c}^+(\mathbf{x})^t \mathbf{h}^{j(\mathbf{x})} \quad \forall \mathbf{x} \in Y$. Let $\mathbf{z}_{(j(\mathbf{x}), \mathbf{x})} = 1$. We have that $(\delta, H, \mathbf{s}, \mathbf{z})$ is a feasible solution for $S2(Y)$ •.
- (iii) Since (i) and (ii) are valid and δ is the value of the objective function for both problems we have that $v(S(Y)) = v(S2(Y))$.

We derived problems to compute upper and lower bounds to $v(MSR(k))$. The S-Q algorithm to be presented iterates between $S2(Y)$ to obtain a lower bound and $Q2(H)$ to obtain an upper bound. Each time we solve $S2(Y)$ we obtain a new lower bound and a new H . Each time we solve $Q2(H)$ we update the best upper bound and obtain a new $\mathbf{y} \in X$ to be added to Y . Since X is a finite set the algorithm is finite. Now we present the S-Q algorithm and Lemma 5 to prove its finitude.

The S-Q algorithm to solve the MSR(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 $v(Q(H^*)) - v(MSR(k)) \leq \epsilon$.

1. Solve $Q2(H)$.
Let (σ, \mathbf{y}) be an optimal solution and let $UB = \min\{UB, v(Q2(H))\}$.
2. If $UB - LB \leq \epsilon$ let $H^* = H$ and stop, otherwise let $Y = Y \cup \{\mathbf{y}\}$.
3. Solve $S2(Y)$.
Let $(\delta, H, \mathbf{s}, \mathbf{z})$ be an optimal solution and let $LB = \delta$.
4. If $UB - LB \leq \epsilon$ let $H^* = H$ and stop, otherwise return to Step 1.

□

Lemma 5 Algorithm S-Q finds an ϵ -optimal solution for the MSR(k) problem in a finite number of steps.

Proof Let $(\delta, H, \mathbf{s}, \mathbf{z})$ be an optimal solution for $S2(Y)$ and let (σ, \mathbf{y}) be an optimal solution for $Q2(H)$. If $\mathbf{y} \in Y$ then:

$$\begin{aligned}
 LB &\stackrel{(1)}{=} \delta \stackrel{(2)}{\geq} \min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{y})^t \mathbf{h} - \mathbf{L}^t \mathbf{y} \stackrel{(3)}{=} v(Q2(H)) \stackrel{(4)}{\geq} UB \\
 &\stackrel{(5)}{\geq} v(MSR(k)) \stackrel{(6)}{\geq} v(S(Y)) \stackrel{(7)}{=} v(S2(Y)) \stackrel{(8)}{=} \delta \stackrel{(9)}{=} LB
 \end{aligned}$$

where

- (1) see Step 3.
- (2) since $(\delta, H, \mathbf{s}, \mathbf{z})$ is a feasible solution for $S2(Y)$ and $\mathbf{y} \in Y$ we have that

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

- (3) By Lemma 1 since (σ, \mathbf{y}) is an optimal solution for $Q2(H)$.
- (4) See Step 1.
- (5) By the definition of MSR(k) since $UB = v(Q2(H)) = v(Q(H))$ for some H with $H \subseteq X$ and $|H| = k$.
- (6) From Lemma 3.

- (7) From Lemma 4.
- (8) Since $(\delta, H, \mathbf{s}, \mathbf{z})$ is an optimal solution for $S2(Y)$.
- (9) See Step 3.

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 becomes a known classic algorithm to find an absolute robust solution (Aissi et al. 2009). Also, if $\epsilon = 0$ the algorithm finds an optimal solution.

Solving the MSR(k) problem may be a task that consumes a lot of computing time and a lot of memory so we consider two greedy approaches to be presented in the next section. □

3 Greedy approaches for the MSR(k) problem

In this section, we present two greedy algorithms for the MSR(k) problem. The first one (the T greedy algorithm) works as follows: let $H \subseteq X$, a problem $T(H)$ in \mathbf{x} is solved to minimize the regret of $H \cup \{\mathbf{x}\}$. If \mathbf{x} is optimal to $T(H)$ then \mathbf{x} is added to H . We solve $T(H)$ iteratively until $|H| = k$. The second one (the Q greedy algorithm) is a very simple application of the first k steps of a multiparametric algorithm to solve $P(c)$ for all $c \in \Omega$. Some lemmas and auxiliary problems are necessary.

3.1 A greedy algorithm for the MSR(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:

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

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 MSR(k) problem

Let $H \subseteq X$.

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 different 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 2000).

\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 $\min_{\mathbf{x} \in \hat{X}} \mathbf{c}^t \mathbf{x} = v(P(\mathbf{c}))$ 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 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. For each $\mathbf{x} \in X$ then $v(Q(H \cup \{\mathbf{x}\}))$ is an upper bound for $v(T(H))$. Let $Y \subseteq X$. We present below a problem to obtain a lower bound named $W(H, Y)$. To do that we reformulate $T(H)$ by using a level set transformation. In the reformulation we have a constraint for each $\mathbf{x} \in X$. Then if $\mathbf{x} \notin Y$ we delete the constraint and we have a relaxation which is a lower bound. Problem $W(H, Y)$ is presented in Lemma 6.

Lemma 6 *Let $H \subseteq X$ with $|H| \geq 1$ and let $Y \subseteq X$ with $|Y| \geq 1$. Let $W(H, Y)$ be a problem in (λ, \mathbf{x}) defined as follows:*

$$\min_{\lambda \in \mathbb{R}, \mathbf{x} \in X} \{ \lambda : \lambda \geq \min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{y})^t \mathbf{h} - \mathbf{L}^t \mathbf{y} \quad \vee \quad \lambda \geq \mathbf{c}^+(\mathbf{y})^t \mathbf{x} - \mathbf{L}^t \mathbf{y} \quad \forall \mathbf{y} \in Y \}$$

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

Proof

$$\begin{aligned} v(T(H)) &\stackrel{(1)}{=} \min_{\mathbf{x} \in X} \max_{\mathbf{y} \in X} \min_{\mathbf{h} \in H \cup \{\mathbf{x}\}} \mathbf{c}^+(\mathbf{y})^t \mathbf{h} - \mathbf{L}^t \mathbf{y} \stackrel{(2)}{\geq} \\ &\min_{\lambda \in \mathbb{R}, \mathbf{x} \in X} \{ \lambda : \lambda \geq \min_{\mathbf{h} \in H \cup \{\mathbf{x}\}} \mathbf{c}^+(\mathbf{y})^t \mathbf{h} - \mathbf{L}^t \mathbf{y} \quad \forall \mathbf{y} \in Y \} \stackrel{(3)}{=} \\ &\min_{\lambda \in \mathbb{R}, \mathbf{x} \in X} \{ \lambda : \lambda \geq \min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{y})^t \mathbf{h} - \mathbf{L}^t \mathbf{y} \quad \vee \quad \lambda \geq \mathbf{c}^+(\mathbf{y})^t \mathbf{x} - \mathbf{L}^t \mathbf{y} \quad \forall \mathbf{y} \in Y \} = \\ &v(W(H, Y)) \end{aligned}$$

where

- (1) From Lemma 1 and the definition of $T(H)$.
- (2) Since $Y \subseteq X$ and by level set transformation.
- (3) A trivial equality.

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

$$\begin{aligned} (W2(H, Y)) \quad &\min \lambda \quad s.t. \\ &\lambda \geq \mathbf{s}_y - \mathbf{L}^t \mathbf{y}, \\ &\mathbf{s}_y \geq \min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{y})^t \mathbf{h} \mathbf{z}_y \quad \forall \mathbf{y} \in Y, \\ &\mathbf{s}_y \geq \mathbf{c}^+(\mathbf{y})^t \mathbf{x} - M_y \mathbf{z}_y \quad \forall \mathbf{y} \in Y, \\ &\mathbf{x} \in X, \quad \mathbf{z}_y \in \{0, 1\} \quad \forall \mathbf{y} \in Y, \quad \lambda \in \mathbb{R}, \quad \mathbf{s}_y \in \mathbb{R} \quad \forall \mathbf{y} \in Y \end{aligned}$$

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

Note that if $\mathbf{z}_y = 1$ then $\lambda \geq \min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{y})^t \mathbf{h} - \mathbf{L}^t \mathbf{y}$. If $\mathbf{z}_y = 0$ then $\lambda \geq \mathbf{c}^+(\mathbf{y})^t \mathbf{x} - \mathbf{L}^t \mathbf{y}$.

Now we have problems to compute upper and lower bounds to $v(T(H))$. The W-Q algorithm to be presented iterates between $W(H, Y)$ to obtain a lower bound and $Q2(H \cup \{x\})$ to obtain an upper bound. Each time we solve $W(H, Y)$ we obtain a new lower bound and a new x . Each time we solve $Q2(H \cup \{x\})$ we update the best upper bound and obtain a new $y \in X$ to be added to Y . Since X is a finite set the algorithm is finite. Now we present the W-Q algorithm and Lemma 7 to prove its finitude.

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

Let $\epsilon \geq 0$. Let $H \subseteq X$ with $|H| \geq 1$. Let $Y \subseteq X$. Let $x \in X$. Let $UB = \infty$. Let $LB = 0$. The output is x^* with $v(Q(H \cup \{x^*\})) - v(T(H)) \leq \epsilon$.

1. Solve $Q2(H \cup \{x\})$. Let (σ, y) be an optimal solution and let $UB = \min\{UB, v(Q2(H \cup \{x\}))\}$.
2. If $UB - LB \leq \epsilon$ let $x^* = x$ and Stop, otherwise let $Y = Y \cup \{y\}$.
3. Solve $W2(H, Y)$ and let (λ, x, s, z) be an optimal solution. Let $LB = \lambda$.
4. If $UB - LB \leq \epsilon$ let $x^* = x$ and Stop, otherwise return to Step 1.

□

Lemma 7 *Let $H \subseteq X$ with $|H| \geq 1$. Algorithm W-Q finds an ϵ -optimal solution for $T(H)$ in a finite number of steps.*

Proof Let (λ, x, s, z) be an optimal solution for $W2(H, Y)$ and let (σ, y) be an optimal solution for $Q2(H \cup \{x\})$. If $y \in Y$ then:

$$\begin{aligned}
 LB &\stackrel{(1)}{=} \lambda \stackrel{(2)}{\geq} \min_{h \in H} \{c^+(y)^t h, c^+(y)^t x\} - L^t y \stackrel{(3)}{=} v(Q2(H \cup \{x\})) \\
 &\stackrel{(4)}{\geq} UB \stackrel{(5)}{\geq} v(T(H)) \stackrel{(6)}{\geq} v(W(H, Y)) \stackrel{(7)}{=} v(W2(H, Y)) \stackrel{(8)}{=} \lambda \stackrel{(9)}{=} LB
 \end{aligned}$$

where

- (1) See Step 3.
- (2) Since (λ, x, s, z) is an optimal solution for $W2(H, Y)$ and $y \in Y$.
- (3) Since (σ, y) is an optimal solution for $Q2(H \cup \{x\})$
- (4) See Step 1.
- (5) From the definition of $T(H)$ and since $UB = v(Q2(H \cup x))$ for some $x \in X$.
- (6) From Lemma 6.
- (7) Since $W2(H, Y)$ is the reformulation of $W(H, Y)$.
- (8) Since (λ, x, s, z) is an optimal solution for $W2(H, Y)$.
- (9) See Step 3.

Since X is a finite set then $UB - LB \leq \epsilon$ in a finite number of steps. Also, if $\epsilon = 0$ the algorithm finds an optimal solution. □

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

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

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

1. Solve $Q2(H)$. Let (σ, \mathbf{x}) be an optimal solution.
2. If $v(Q2(H)) = 0$ 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 (σ^*, \mathbf{x}^*) is an optimal solution for $Q2(H)$ then $\mathbf{c}^+(\mathbf{x}^*)$ is the worst scenario for H and \mathbf{x}^* is chosen 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 than using the Q-greedy algorithm. The computational results presented in Sect. 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 MrelSR(k) problem

In this section we present an algorithm to obtain an optimal solution for the MrelSR(k) problem.

Our presentation follows the same scheme that we used for the MSR(k) problem and the motivation and proof of some lemmas and algorithms are analogous. Thus, for the sake of simplicity and in order to save space, we omit some motivation and proofs.

If $H \subseteq X$ with $|H| = k$ then $v(Q_r(H))$ is an upper bound of $v(\text{MrelSR}(k))$. Let $Y \subseteq X$. We present later a problem $S_r(Y)$ to obtain a lower bound. The algorithm iterates between solving the problems $S_r(Y)$ (to generate a new H) and $Q_r(H)$ (to generate a new \mathbf{y} to be added to Y) until the gap is closed. Some lemmas and auxiliary problems are necessary.

Lemma 8 *Let $H \subseteq X$ with $|H| \geq 1$. Let $QX_r(H)$ be a problem in (\mathbf{x}) defined as follows:*

$$\max_{\mathbf{x} \in X} \frac{\min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{x})^t \mathbf{h} - \mathbf{c}^+(\mathbf{x})^t \mathbf{x}}{\mathbf{c}^+(\mathbf{x})^t \mathbf{x}}$$

then:

- (i) $v(Q_r(H)) = v(QX_r(H))$
- (ii) *If \mathbf{c}^* is an optimal solution for $Q_r(H)$ and \mathbf{x}^* is an optimal solution for $P(\mathbf{c}^*)$ then \mathbf{x}^* is an optimal solution for $QX_r(H)$.*

(iii) If \mathbf{x}^* is an optimal solution for $QX_r(H)$ then $\mathbf{c}^* = \mathbf{c}^+(\mathbf{x}^*)$ is an optimal solution for $Q_r(H)$.

Proof (i) 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:

$$\begin{aligned}
 v(Q_r(H)) &\stackrel{(1)}{=} \frac{\min_{\mathbf{h} \in H} \mathbf{c}^{*t} \mathbf{h} - \mathbf{c}^{*t} \mathbf{x}^*}{\mathbf{c}^{*t} \mathbf{x}^*} \stackrel{(2)}{\leq} \max_{\mathbf{c} \in \Omega, \mathbf{x} \in X} \frac{\min_{\mathbf{h} \in H} \mathbf{c}^t \mathbf{h} - \mathbf{c}^t \mathbf{x}}{\mathbf{c}^t \mathbf{x}} \\
 &\stackrel{(3)}{\leq} \max_{\mathbf{x} \in X} \frac{\min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{x})^t \mathbf{h} - \mathbf{c}^+(\mathbf{x})^t \mathbf{x}}{\mathbf{c}^+(\mathbf{x})^t \mathbf{x}} \stackrel{(4)}{=} v(QX_r(H))
 \end{aligned}$$

where

- (1) From the definition of $Q_r(H)$ and since \mathbf{c}^* is an optimal solution for $Q_r(H)$ and \mathbf{x}^* is an optimal solution for $P(\mathbf{c}^*)$.
- (2) a trivial inequality.
- (3) Since $\mathbf{c}^t \mathbf{h} - \mathbf{c}^t \mathbf{x} \leq \mathbf{c}^+(\mathbf{x})^t \mathbf{h} - \mathbf{c}^+(\mathbf{x})^t \mathbf{x}$ for all $\mathbf{x} \in X$, for all $\mathbf{c} \in \Omega$ and for all $\mathbf{h} \in X$ and $\mathbf{c}^t \mathbf{x} \geq \mathbf{c}^+(\mathbf{x})^t \mathbf{x}$ for all $\mathbf{x} \in X$ and for all $\mathbf{c} \in \Omega$.
- (4) From the definition of $QX_r(H)$.

Let \mathbf{x}^* be an optimal solution for $QX_r(H)$ then we have:

$$\begin{aligned}
 v(QX_r(H)) &\stackrel{(5)}{=} \frac{\min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{x}^*)^t \mathbf{h} - \mathbf{c}^+(\mathbf{x}^*)^t \mathbf{x}^*}{\mathbf{c}^+(\mathbf{x}^*)^t \mathbf{x}^*} \stackrel{(6)}{\leq} \frac{\min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{x}^*)^t \mathbf{h} - v(P(\mathbf{c}^+(\mathbf{x}^*)))}{v(P(\mathbf{c}^+(\mathbf{x}^*)))} \\
 &\stackrel{(7)}{\leq} \max_{\mathbf{c} \in \Omega} \frac{\min_{\mathbf{h} \in H} \mathbf{c}^t \mathbf{h} - v(P(\mathbf{c}))}{v(P(\mathbf{c}))} \stackrel{(8)}{=} v(Q_r(H))
 \end{aligned}$$

where

- (5) From the definition of $QX_r(H)$.
- (6) Since $v(P(\mathbf{c}^+(\mathbf{x}^*))) \leq \mathbf{c}^+(\mathbf{x}^*)^t \mathbf{x}^*$
- (7) A trivial inequality.
- (8) From the definition of $Q_r(H)$.

(ii) Let \mathbf{c}^* be an optimal solution for $Q_r(H)$ and let \mathbf{x}^* be an optimal solution for $P(\mathbf{c}^*)$, then

$$\begin{aligned}
 v(QX_r(H)) &\stackrel{(1)}{=} v(Q_r(H)) \stackrel{(2)}{=} \frac{\min_{\mathbf{h} \in H} \mathbf{c}^{*t} \mathbf{h} - v(P(\mathbf{c}^*))}{v(P(\mathbf{c}^*))} \stackrel{(3)}{=} \frac{\min_{\mathbf{h} \in H} \mathbf{c}^{*t} \mathbf{h} - \mathbf{c}^{*t} \mathbf{x}^*}{\mathbf{c}^{*t} \mathbf{x}^*} \\
 &\stackrel{(4)}{\leq} \frac{\min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{x}^*)^t \mathbf{h} - \mathbf{c}^+(\mathbf{x}^*)^t \mathbf{x}^*}{\mathbf{c}^+(\mathbf{x}^*)^t \mathbf{x}^*} \stackrel{(5)}{\leq} v(QX_r(H))
 \end{aligned}$$

therefore \mathbf{x}^* is an optimal solution for $QX_r(H)$,

where

- (1) From (i).
- (2) Since \mathbf{c}^* is an optimal solution for $Q_r(H)$.
- (3) Since \mathbf{x}^* is an optimal solution for $P(\mathbf{c}^*)$
- (4) Since $\mathbf{c}^{*t}\mathbf{h} - \mathbf{c}^{*t}\mathbf{x}^* \leq \mathbf{c}^+(\mathbf{x}^*)^t\mathbf{h} - \mathbf{c}^+(\mathbf{x}^*)^t\mathbf{x}^*$ and $\mathbf{c}^+(\mathbf{x}^*)^t\mathbf{x}^* = \mathbf{L}^t\mathbf{x}^* \leq \mathbf{c}^{*t}\mathbf{x}^*$
- (5) From the definition of $QX_r(H)$.

(iii) Let \mathbf{x}^* be an optimal solution for $QX_r(H)$ and let $\mathbf{c}^* = \mathbf{c}^+(\mathbf{x}^*)$.

$$\begin{aligned}
 v(Q_r(H)) &\stackrel{(1)}{=} v(QX_r(H)) \stackrel{(2)}{=} \frac{\min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{x}^*)^t\mathbf{h} - \mathbf{c}^+(\mathbf{x}^*)^t\mathbf{x}^*}{\mathbf{c}^+(\mathbf{x}^*)^t\mathbf{x}^*} \stackrel{(3)}{=} \frac{\min_{\mathbf{h} \in H} \mathbf{c}^{*t}\mathbf{h} - \mathbf{c}^{*t}\mathbf{x}^*}{\mathbf{c}^{*t}\mathbf{x}^*} \\
 &\stackrel{(4)}{\leq} \frac{\min_{\mathbf{h} \in H} \mathbf{c}^{*t}\mathbf{h} - v(P(\mathbf{c}^*))}{v(P(\mathbf{c}^*))} \stackrel{(5)}{\leq} v(Q_r(H))
 \end{aligned}$$

therefore \mathbf{c}^* is an optimal solution for $Q_r(H)$, where

- (1) From (i).
- (2) Since \mathbf{x}^* is an optimal solution for $QX_r(H)$.
- (3) Since $\mathbf{c}^+(\mathbf{x}^*) = \mathbf{c}^*$.
- (4) Since $v(P(\mathbf{c}^*)) \leq \mathbf{c}^{*t}\mathbf{x}^*$.
- (5) From the definition of $Q_r(H)$ •

$QX_r(H)$ is a combinatorial optimization problem with a rational objective function (Megiddo 1979). Hence, we know some basic properties as follows (for the sake of completeness we present a proof): □

Proposition 2 *Let $k \geq 1$, let $H \subseteq X$ with $|H| = k$, let $\mu \geq 1$ and let $RX_r(\mu, H)$ be a problem in (\mathbf{x}) defined as:*

$$(RX_r(\mu, H)) \max_{\mathbf{x} \in X} \min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{x})^t\mathbf{h} - \mu\mathbf{c}^+(\mathbf{x})^t\mathbf{x}$$

then:

- (i) $v(RX_r(\mu, H)) = 0$ if and only if $\mu = v(QX_r(H)) + 1$.
- (ii) If $v(RX_r(\mu, H)) = 0$ and \mathbf{x}^* is an optimal solution for $RX_r(\mu, H)$ then \mathbf{x}^* is an optimal solution for $QX_r(H)$.
- (iii) $v(RX_r(\mu, H))$ is a piecewise linear and decreasing convex function in μ .

Proof See ‘‘Appendix B’’.

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

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

$$\max_{\phi \in \mathbb{R}, \mathbf{x} \in X} \{\phi - \mu\mathbf{L}^t\mathbf{x} : \phi \leq \mathbf{c}^+(\mathbf{x})^t\mathbf{h} \ \forall \mathbf{h} \in H\}$$

Since X is a finite set and $v(RX_r(\mu, H))$ is a piecewise linear and decreasing convex function in μ we have that the next algorithm finds μ^* with $v(RX(\mu^*, H)) = 0$ in a finite number of steps. For the sake of completeness we present a proof. \square

Algorithm Find- μ^*

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

1. Solve $RX_r(\mu, H)$. Let \mathbf{x} be an optimal solution.
2. If $v(RX_r(\mu, H)) = 0$ let $\mu^* = \mu$, let $\mathbf{x}^* = \mathbf{x}$ and stop.
3. Let $\mu = \frac{\min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{x})^t \mathbf{h}}{\mathbf{c}^+(\mathbf{x})^t \mathbf{x}}$ and return to Step 1.

Proposition 3 Algorithm Find- μ^* finds μ^* with $v(RX_r(\mu^*, H)) = 0$ and \mathbf{x}^* optimal for $QX_r(H)$ in a finite number of steps.

Proof See ‘‘Appendix C’’. \square

Lemma 9 Let $Y \subseteq X$ with $|Y| \geq 1$. Let $S_r(Y)$ be a problem in (σ_r, H) defined as follows:

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

$$\text{then } v(MrelSR(k)) \geq v(S_r(Y))$$

We omit the proof.

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

$$\begin{aligned} (S_r(Y)) \quad & \min \sigma_r \quad s.t. \\ & \mathbf{L}^t \mathbf{x} \sigma_r \geq \mathbf{s}_x - \mathbf{L}^t \mathbf{x} \quad \forall \mathbf{x} \in Y, \quad \forall j \in [k], \\ & \mathbf{s}_x \geq \mathbf{c}^+(\mathbf{x})^t \mathbf{h}^j - (1 - \mathbf{z}_{(j,\mathbf{x})}) M_x \quad \forall \mathbf{x} \in Y, \\ & \sum_{j=1}^k \mathbf{z}_{(j,\mathbf{x})} = 1 \quad \forall \mathbf{x} \in Y, \\ & \mathbf{h}^{(j)} \in X \quad \forall j \in [k], \quad \mathbf{z}_{(j,\mathbf{x})} \in \{0, 1\} \quad \forall \mathbf{x} \in Y \quad \forall j \in [k], \quad \sigma_r \in \mathbb{R}, \quad \mathbf{s}_x \in \mathbb{R} \quad \forall \mathbf{x} \in Y \end{aligned}$$

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

Sr-Qr algorithm to solve the MrelSR(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 $v(QX_r(H^*)) - v(MrelSR(k)) \leq \epsilon$.

1. Solve $QX_r(H)$. Let \mathbf{y} be an optimal solution and let $UB = \min\{UB, v(Q_r(H))\}$.
2. If $UB - LB \leq \epsilon$ let $H^* = H$ and stop, otherwise let $Y = Y \cup \{\mathbf{y}\}$.
3. Solve $SX_r(Y)$. Let $(\sigma_r, H, \mathbf{s}, \mathbf{z})$ be an optimal solution and let $LB = \sigma_r$.

4. If $UB - LB \leq \epsilon$ let $H^* = H$ and stop, otherwise return to Step 1

Lemma 10 *Algorithm Sr-Qr finds an ϵ -optimal solution for MrelSR(k) in a finite number of steps.*

We omit the proof.

5 Greedy approaches for the MrelSR(k) problem

In this section we present two greedy algorithms for the MrelSR(k) problem.

Our presentation follows the same scheme that we used for the greedy approach to MSR(k) problem and the motivation and proof of some lemmas and algorithms are analogous. Thus, for the sake of simplicity and in order to save space, we omit some motivation and proofs.

The first one (the T_r -greedy algorithm) works as follows: let $H \subseteq X$, a problem $T_r(H)$ in \mathbf{x} is solved to minimize the relative regret of $H \cup \{\mathbf{x}\}$. If \mathbf{x} is optimal to $T_r(H)$ then \mathbf{x} is added to H . We solve $T_r(H)$ iteratively until $|H| = k$. The second one (the Q_r -greedy algorithm) is analogous to the Q greedy algorithm. Some lemmas and auxiliary problems are necessary.

5.1 A greedy algorithm for the MrelSR(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:

$$(T_r(H)) \min_{\mathbf{x} \in X} v(Q_r(H \cup \{\mathbf{x}\}))$$

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 T_r -greedy algorithm for the MrelSR(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 T_r -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 different solution even if $|H| < k$.

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

Lemma 11 Let $Y \subseteq X$ with $|Y| \geq 1$. Let $W_r(H, Y)$ be a problem in (σ_W, \mathbf{x}) defined as follows:

$$\begin{aligned} \min_{\sigma_W \in \mathbb{R}, \mathbf{x} \in X} \{ & \sigma_W : \mathbf{L}^t \mathbf{y} \sigma_W \geq \min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{y})^t \mathbf{h} - \mathbf{L}^t \mathbf{y} \quad \forall \\ & \mathbf{L}^t \mathbf{y} \sigma_W \geq \mathbf{c}^+(\mathbf{y})^t \mathbf{x} - \mathbf{L}^t \mathbf{y} \quad \forall \mathbf{y} \in Y \} \end{aligned}$$

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

We omit the proof.

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 $(\sigma_W, \mathbf{x}, \mathbf{s}, \mathbf{z})$ as follows:

$$\begin{aligned} (W_r(H, Y)) \quad & \min \quad \sigma_W \quad s.t \\ & \mathbf{L}^t \mathbf{y} \sigma_W \geq \mathbf{s}_y - \mathbf{L}^t \mathbf{y}, \\ & \mathbf{s}_y \geq \min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{y})^t \mathbf{h} \mathbf{z}_y \quad \forall \mathbf{y} \in Y, \\ & \mathbf{s}_y \geq \mathbf{c}^+(\mathbf{y})^t \mathbf{x} - M_y \mathbf{z}_y \quad \forall \mathbf{y} \in Y, \\ & \mathbf{x} \in X, \quad \mathbf{z}_y \in \{0, 1\} \quad \forall \mathbf{y} \in Y, \quad \sigma_W \in \mathbb{R}, \quad \mathbf{s}_y \in \mathbb{R} \quad \forall \mathbf{y} \in Y \end{aligned}$$

and we use $M_y = \max_{\mathbf{x} \in X} \mathbf{c}^+(\mathbf{y})^t \mathbf{x}$ 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 $UB - LB \leq \epsilon$ let $\mathbf{x}^* = \mathbf{x}$ and 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 $UB - LB \leq \epsilon$ let $\mathbf{x}^* = \mathbf{x}$ and stop, otherwise return to Step 1.

Lemma 12 Algorithm *Wr-Qr* finds an ϵ -optimal solution for $T_r(H)$ in a finite number of steps.

We omit the proof.

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

The Qr-greedy algorithm

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

1. Solve $QX_r(H)$. Let \mathbf{x} be an optimal solution.
2. If $v(QX_r(H)) = 0$ 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 $v(Q_r(H \cup x^{(1)})) \geq v(Q_r(H \cup x^{(2)}))$. 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 than 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)Core(TM) i7-3630QM CPU, HP envy dv6, with 12.00 GB Ram, 2.40 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 all the parameters of CPLEX in their default values with the exemption of the tolerance to declare an integer value that was changed to 10^{-12} for the SP problem with euclidean topology to avoid numerical difficulties that appeared by using the default value. We use $\epsilon = 0$.

We use the $*$ algorithm, with $*$ \in {Q-greedy,T-greedy,Qr-greedy,Tr-greedy}, beginning with:

$H = \{h^{(1)}\}$ where $h^{(1)}$ is either an absolute robust solution or a relative robust solution according the case and $Y = \emptyset$.

If $k = 2$ we use the $S - Q$ and $Sr - Qr$ algorithms beginning with H where H is the output of the T-greedy (Tr-greedy) algorithm used for $k = 2$ and $Y = \emptyset$.

If $k = 3$ we use the $S - Q$ and $Sr - Qr$ algorithms beginning with H where H is the output of the T-greedy (Tr-greedy) algorithm used for $k = 3$ and $Y = \emptyset$.

In order to solve $T(H)$ ($T_r(H)$) by using the W-Q (Wr-Qr) algorithm the initial x is an optimal solution for $Q2(H)$ ($QX_r(H)$).

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 ($*$ \in {Q-greedy,T-greedy,Qr-greedy,Tr-greedy,S-Q,Sr-Qr}):

- k : the number of solutions to define H ,
- t^* : the largest time in seconds to generate H where $|H| = k$ with the $*$ algorithm. The time to obtain the first H is included,
- $avet^*$: the times average for algorithm $*$,
- it^* : the largest number of iterations (the iterations to obtain the first H is included) with the $*$ algorithm,
- red^*k is the reduction percentage for the regret values if we use the set H (with k solutions) generated by the algorithm $*$ instead of the absolute(relative) robust solution (the values presented in the tables are the average for the set considered),
- $difk$ and $difrk$ (with $k \in \{2, 3\}$) are the differences of the average values of the reduction percentage for the regret values between the optimal solutions

and the solutions generated with the T-greedy algorithms as follows: dif2=redS-Q2-redT2,dif3=redS-Q3-redT3, dif2r=redSr-Qr2 - redTr2 and dif3r=redSr-Qr3 - redTr3,

- f: the number of times that the S-Q(Sr-Qr) algorithm failed to obtain an optimal solution with a time limit equal to 3600 s (without including the time to obtain $h^{(1)}$ and the initial H according the case).

Some remarks are necessary: If $k \in \{2, 3\}$ we use a time limit equal to 3600 s for the S-Q and Sr-Qr algorithms (without including the time to obtain $h^{(1)}$ and the initial H according the case), that implies that the overall time may be greater than 3600. There is no time limit for the heuristic algorithms for $2 \leq k \leq 6$. If the S-Q (Sr-Qr) algorithm failed to obtain an optimal solution we use the best lower bound generated to compute the differences between optimal solutions and the solutions generated by the greedy algorithms. If a problem was not solved the execution time and the iterations are not included to compute averages and worst cases.

6.2 Data generation

At the present time we have computational results for the Shortest Path (SP) and the p -Medians (p -M) problems.

6.2.1 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} to \hat{w} with the minimum 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 $x_e = 1$ if the edge e is selected.

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

$$\begin{aligned}
 X = \{ \mathbf{x} : & \\
 & \sum_{e \in E^+(\hat{v})} x_e = 1, \\
 & \sum_{e \in E^-(v)} x_e - \sum_{e \in E^+(v)} x_e = 0 \quad \forall v \in V \text{ with } v \notin \{\hat{v}, \hat{w}\}, \\
 & \sum_{e \in E^-(\hat{w})} x_e = 1, \\
 & \mathbf{x} \in \{0, 1\}^{|E|} \}
 \end{aligned}$$

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

$$(P(\mathbf{c})) \min_{\mathbf{x} \in X} \mathbf{c}'\mathbf{x}$$

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

We define G with two different topologies: A Mesh topology and an Euclidean topology as follows:

Mesh topology

We use $G = (V, E)$ with a simple mesh topology: there is only one node at level 0, named $(0, 1)$. Next we have m levels with r nodes at each level. Node $(0, 1)$ is connected with all the nodes of level 1: from $(1, 1)$ to $(1, r)$. Every node of level i is connected with all nodes of level $i + 1$ (with $i \in \{1, \dots, m\}$). That is: node (i, k) (with $k \in \{1, \dots, r\}$) is connected with nodes $(i + 1, 1), \dots, (i + 1, r)$. Finally, the nodes of level m are connected with the only node at level $m + 1$ named $(m + 1, 1)$.

We have $|V| = 2 + rm$ and $|E| = 2r + r^2(m - 1)$. We use $\hat{v} = (0, 1)$ and $\hat{w} = (m + 1, 1)$. Every path from \hat{v} to \hat{w} has $m + 1$ edges.

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

Euclidean topology

The graphs are generated following Hanasusanto et al. (2015) as follows:

In each problem instance, the nodes correspond to $|V|$ points with coordinates that are chosen uniformly at random in the square $[0, 10] \times [0, 10]$. We choose the pair of nodes with the largest Euclidean distance as \hat{v} and \hat{w} (the start and terminal nodes). Let $0 < p < 1$. To generate E , we begin with a fully connected graph and remove $100 \times p\%$ of the arcs in order of decreasing Euclidean distance, that is, starting with the longest arcs. If the graph is not connected we do not use it. The data were generated as follows: for each $e \in E$ the travel time between each pair of adjacent nodes varies between 100% and 150% of the Euclidean distance between the nodes.

6.2.2 Data for the p -Medians problems

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

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

Let $w_{ij} \in \{0, 1\}$ ($i, j \in J$) with $w_{ij} = 1$ if and only if the demand location j is assigned to a median located at i .

Let q_{ij} such that $LL_{ij} \leq q_{ij} \leq UU_{ij}$ for all $i, j \in J$. The p -M problem with cost q is a problem in w, z defined as follows:

$$\begin{aligned} \min \sum_{i \in J} \sum_{j \in J} q_{ij} w_{ij} \quad s.t. \\ w_{ij} \leq z_i \quad \forall i, j \in J \end{aligned}$$

$$\begin{aligned} \sum_{i \in J} z_i &= p \\ \sum_{i \in J} w_{ij} &= 1 \quad \forall j \in J \\ z_i \in \{0, 1\}, \quad w_{ij} &\in \{0, 1\} \quad \forall i, j \in J \end{aligned}$$

Let $n = s + s^2$. With very simple variable changes the p -M problem may be rewritten as $\min_{x \in X} \mathbf{c}'\mathbf{x}$ where $X \subseteq \{0, 1\}^n$ and $\mathbf{L} \leq \mathbf{c} \leq \mathbf{U}$ for appropriate vectors $\mathbf{L}, \mathbf{U} \in \mathbb{R}^n$.

Since X may be 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 to location j computed as $\mathbf{d}_{ij} = |\mathbf{x1}_i - \mathbf{x1}_j| + |\mathbf{x2}_i - \mathbf{x2}_j|$. Let $\gamma > 0$. Let $\mathbf{LL}_{ij} = \mathbf{d}_{ij}\mathbf{D}_j$ and let $\mathbf{UU}_{ij} = (1 + r_{ij}\gamma)\mathbf{LL}_{ij}$ where r_{ij} is taken from $U(0, 1)$.

6.3 Performance of the algorithms

6.3.1 Absolute case (MSR(k))

Tables

Results associated with the use of the S-Q and T-Greedy algorithms for p -M and SP problems (with a mesh topology and a euclidean topology) with $k \in \{2, 3\}$ may be seen in Table 1 (120 problems were considered). For each parameter combination 10 problems were solved.

Results associated with the use of the Q-greedy and T-Greedy algorithms with $k \in \{2, \dots, 6\}$ may be seen in Table 2 (170 problems were considered). We can see times for $k = 6$ and the percentages of reduction of the regret values associated with the two algorithms.

Complimentary details may be seen in Tables 4, 5, 6, 7, 8 and 9 (see ‘‘Appendix D’’, 290 problems were considered). We can see times, iterations and failures of the optimal algorithm for the time limit considered. Also, we can see percentages of reduction of the regret values and the difference between optimal and greedy algorithms.

Remarks We present in Table 1 the times average for algorithm S-Q (avetS-Q) and the differences of the average values of the reduction percentage for the regret values between the optimal solutions and the solutions generated with the T-greedy algorithms (difk wit $k \in \{2, 3\}$).

S-Q algorithm failed 3 times to obtain an optimal solution for p -M problems with $(m, p, \gamma, k) = (60, 5, 0.15, 3)$ and for that cases the execution times are not included to compute the average. None of problems were solved with $(m, p, \gamma, k) = (60, 5, 0.35, 3)$. When the S-Q algorithm fails we use the best lower bound generated

to compute the differences between optimal solutions and the solutions generated by the T-greedy algorithm.

From Table 1 it is evident that when the dimensions and the size of the uncertainty set increase, the computational effort increases.

See the columns with $\gamma = 0.15$ and $\gamma = 0.35$ with the rest of parameters fixed for SP problems with mesh topology and p -M problems to note that the time increases in general and sometimes considerably (for example for the SP problems with $(r, m, \gamma, k) = (10, 50, 0.15, 3)$ and $(r, m, \gamma, k) = (10, 50, 0.35, 3)$ we have 53.30 and 256.62 seconds for the time averages).

With (γ, k) fixed move from one line to another with larger dimensions to note that the time increases in general (for example for the p -M problems with $(m, p, \gamma, k) = (60, 2, 0.15, 2)$ and $(m, p, \gamma, k) = (60, 5, 0.15, 2)$ we have 118.45 and 563.49 seconds for the time averages). Also, with (p, k) fixed for the SP problems with euclidean topology see the times as $|V|$ increases (for example if $(p, k) = (0.70, 3)$ the average times are 9.03, 18.59, 72.90 and 698.33 seconds for $|V| = 15, 20, 25, 30$).

For the problems included in Tables 1, 4, 5, 6, 7, 8 and 9 the T-Greedy algorithm found near optimal solutions. For $k = 2$, the worst case for the average difference is 1.52% for the SP problems with the mesh topology. For $k = 3$, the worst case for the average difference is 6% for the p -M problems.

For the problems included in Tables 1, 4, 5, 6, 7, 8 and 9 the S-Q algorithm was used to solve 290 problems and found an optimal solution 287 times for $k = 2$ and 241 times for $k = 3$ with a time limit of 3600 s. The worst case was for the set with $(p, m, \gamma, k) = (60, 5, 0.35, 3)$ (p -M problems), in which none of the ten problems included were solved.

It is evident (see Table 2) that as we can expect the reduction percentages and the computational effort of the heuristic algorithms are larger, in general, using the T-Greedy algorithm.

The use of the greedy approaches is justified because we can find near optimal solutions with a tolerable computational effort. The percentages of reduction obtained are, in general, large with some few exemptions.

6.3.2 Relative case ($MrelSR(k)$)

Tables

Results associated with the use of the Qr-greedy and Tr-Greedy algorithms with $k \in \{2, \dots, 6\}$ may be seen in Table 3 (150 problems). We can see times and the percentages of reduction of the regret values associated with the two algorithms.

Complimentary details may be seen in Tables 10, 11, 12, 13, 14 and 15 (see “Appendix D”). Results associated with the use of the Sr-Qr, Qr-Greedy and Tr-Greedy algorithms for p -M and SP problems (with a mesh topology and a euclidean topology) with $k \in \{2, 3\}$ are presented. We can see times, iterations and failures of the optimal algorithm for the time limit considered for 180 problems. Also, we can see percentages of reduction of the regret values and the difference between optimal and greedy algorithms.

Table 1 (i) SP problems with mesh topology and parameters (r, m, γ) , (ii) p -M problems with parameters (m, p, γ) and (iii) SP problems with euclidean topology and parameter $(|V|, p)$, where $\gamma \in \{0.15, 0.35\}$, $p \in \{0.35, 0.70\}$. T-Greedy and S-Q algorithms for the MSR(k) problem where $k \in \{2, 3\}$. 10 problems in each set

		avetS-Q				difk				
		k								
			2	3		2	3			
(i)	r	$m; \gamma$	0.15	0.35	0.15	0.35	0.15	0.35	0.15	0.35
	10	25	4.12	7.68	8.75	22.38	0.00	0.79	0.00	5.16
		50	26.47	542.39	53.30	256.62	0.00	2.99	1.84	9.14
	15	25	8.03	15.03	13.44	41.34	1.32	1.36	4.32	2.18
		50	34.58	159.83	70.17	341.38	0.00	7.42	5.75	7.22
(ii)	m	$p; \gamma$	0.15	0.35	0.15	0.35	0.15	0.35	0.15	0.35
	30	2	15.60	11.82	18.92	25.74	0.05	0.09	7.56	2.44
		5	10.53	25.42	18.82	124.84	3.85	0.83	4.44	7.55
	60	2	118.45	108.08	344.83	643.08	2.49	0.37	7.20	0.75
		5	563.49	546.48	1280.83	–	2.08	0.74	7.18	9.43
(iii)	$ V $	$; p$	0.35	0.70	0.35	0.70	0.35	0.70	0.35	0.70
	15		2.84	3.21	4.79	9.03	0.42	2.45	2.52	3.18
	20		8.79	6.66	162.40	18.59	1.53	2.03	2.67	2.63
	25		9.07	39.35	219.22	72.90	1.02	1.56	0.27	1.35
	30		15.90	11.58	587.11	698.33	0.86	1.21	4.50	3.39

Remarks The percentages of reduction obtained are, in general, large with some few exemptions (see Table 3).

It is evident that as we can expect the reduction percentages and the computational effort of the heuristic algorithms are larger, in general, using the Tr-Greedy algorithm (see Table 3).

For the problems included in Tables 10, 11, 12, 13, 14 and 15 the Qr-Greedy and Tr-Greedy algorithms found near optimal solutions with a tolerable computational effort. For $k = 2$, the worst case for the average difference is 1.52% for the p -M problems. For $k = 3$, the worst case for the average is 5.15% for the SP problem with a mesh topology.

For the problems included in Tables 10, 11, 12, 13, 14 and 15, the Sr-Qr algorithm was used to solve 180 problems and found an optimal solution 179 times for $k = 2$ and 160 times for $k = 3$ with a time limit of 3600 s. In this case, the worst scenario was for p -M problems with $(m, p, \gamma, k) = (80, 5, 0.15, 3)$, the algorithm failed to obtain an optimal solution for 5 problems.

Again, it is evident that as we can expect the computational effort increases, in general, while the dimensions increase. Also, the use of the greedy approaches is justified because we can find near optimal solutions with a tolerable computational effort.

Table 2 (i) SP problems with mesh topology and parameters (r, m, γ) , (ii) p -M problems with parameters (m, p, γ) and (iii) SP problems with euclidean topology and parameters $(|V|, p)$. Q-greedy and T-greedy algorithms for the $MSR(k)$ problem where $k \in \{2, \dots, 6\}$. Reduction percentage for the regret values and times (for $k = 6$). 10 problems in each set

(i)	γ	r	m	*	red*2	red*3	red*4	red*5	red*6	avet*($k = 6$)		
	0.35	20	25	Q	18.70	42.18	50.10	55.96	60.53	6.51		
				T	32.89	46.40	59.83	65.06	71.46	25.04		
				150	Q	11.32	19.39	26.05	30.00	34.49	17.82	
					T	20.60	30.08	35.09	39.36	41.92	82.56	
				75	Q	7.92	13.34	17.67	21.86	27.41	49.29	
					T	17.17	23.53	28.33	33.09	36.49	200.22	
	30	25	Q	25.28	36.48	43.33	49.91	55.38	10.34			
			T	29.11	40.14	44.50	52.18	61.80	41.67			
			50	Q	15.20	24.82	28.45	38.51	46.72	36.11		
				T	21.70	30.32	34.49	46.13	49.34	160.95		
			75	Q	6.69	12.26	17.36	19.89	22.53	103.79		
				T	13.53	21.61	26.02	28.83	32.92	467.07		
(ii)	γ	m	p									
	0.15	100	5	Q	7.02	18.14	20.66	25.22	27.57	99.75		
				T	16.05	24.13	28.01	31.32	33.67	309.91		
				10	Q	9.67	14.80	17.79	20.04	24.30	89.24	
					T	18.38	26.49	30.41	33.75	37.13	341.31	
				200	5	Q	6.21	10.57	13.52	16.56	17.94	795.21
					T	7.30	12.18	15.59	17.37	20.84	3576.43	
	0.35	100	10	Q	11.06	13.57	15.45	16.92	18.02	1797.80		
				T	13.43	18.83	20.98	22.23	23.91	5396.09		
				5	Q	6.65	10.44	13.24	15.24	17.24	150.68	
					T	9.60	14.41	18.23	20.02	21.48	460.95	
				10	Q	6.85	10.10	13.03	14.80	16.28	338.50	
					T	12.98	17.06	19.22	21.23	22.58	1260.21	
200	5	Q	2.86	4.18	6.10	7.25	7.75	1864.16				
		T	3.98	5.80	6.98	8.48	10.28	6517.00				
(iii)	p	V										
	0.35	50	Q	0.69	1.88	3.52	4.74	7.66	2.70			
			T	1.17	2.45	4.17	5.68	7.92	12.56			
			100	Q	0.26	0.55	1.20	1.46	2.00	12.77		
				T	0.31	0.80	1.31	1.67	2.23	77.08		
			0.70	50	Q	2.85	5.97	10.48	12.64	14.88	6.45	
					T	3.45	7.68	10.68	12.11	14.06	21.55	
	100	Q	1.25	2.87	4.16	5.65	6.71	20.30				
		T	1.70	3.32	4.56	5.77	6.47	108.51				

Table 3 (i) SP problems with mesh topology and parameters (r, m, γ) , (ii) p -M problems with parameters (m, p, γ) and (iii) SP problems with euclidean topology and parameters $(|V|, p)$. Qr-greedy and Tr-greedy algorithms for the $MrelSR(k)$ problem where $k \in \{2, \dots, 6\}$. Reduction percentage for the regret values and times (for $k = 6$). 10 problems in each set

(i)	γ	r	m	*	red*2	red*3	red*4	red*5	red*6	avet*($k = 6$)		
	0.35	20	25	Qr	29.19	37.50	53.02	67.14	72.68	11.46		
				Tr	30.72	41.64	57.54	69.51	74.74	39.03		
				50	Qr	25.14	31.92	38.17	43.38	51.89	36.41	
					Tr	32.29	40.07	45.90	50.77	56.75	144.91	
				75	Qr	6.23	12.75	16.07	19.15	21.34	101.20	
					Tr	15.28	23.89	27.93	31.67	33.90	462.34	
	30	25	Qr	44.32	61.58	69.17	81.01	85.51	27.31			
			Tr	53.45	70.05	77.40	81.47	85.71	80.48			
		50	Qr	14.77	29.81	35.64	38.14	43.41	112.74			
			Tr	23.87	36.48	43.73	49.39	54.21	395.65			
		75	Qr	10.96	23.24	28.40	33.89	37.02	178.91			
			Tr	25.16	32.42	40.02	44.56	46.54	908.42			
(ii)	γ	m	p									
	0.15	80	5	Qr	8.04	18.11	25.35	29.45	32.04	62.31		
				Tr	16.51	25.69	31.05	35.31	38.54	235.79		
				100	Qr	8.04	18.11	25.35	29.45	32.04	62.31	
					Tr	16.51	25.69	31.05	35.31	38.54	235.79	
				200	Qr	8.40	13.20	15.19	17.65	19.00	1439.50	
					Tr	11.03	15.86	18.51	20.36	21.71	6068.38	
	0.35	80	Qr	4.68	10.79	12.41	16.49	19.25	130.65			
			Tr	8.49	14.58	18.11	20.75	22.97	501.02			
		100	Qr	6.03	10.97	13.96	16.20	17.95	180.77			
			Tr	8.72	13.00	15.70	17.95	20.49	757.35			
			(iii)	p	V							
			0.35	50	Qr	1.53	4.62	8.02	9.79	11.93	6.02	
Tr	2.16	5.06			8.36	10.06	11.73	23.58				
100	Qr	0.30		1.30	1.62	2.23	2.83	14.95				
	Tr	0.51		1.46	1.85	2.46	3.21	117.23				
0.70	50	Qr		7.69	11.10	17.16	19.94	21.80	10.32			
		Tr		9.66	13.65	16.70	19.23	20.69	40.56			
	100	Qr	1.82	3.88	6.82	9.14	10.97	28.44				
		Tr	2.72	4.63	6.58	8.74	9.90	203.28				

7 Conclusions and further extensions

7.1 Conclusions

We studied the min max min robust (relative) regret problem. We presented the regret and the relative regret of a set with k elements. Algorithms to find a set with the minimum (relative) regret were developed. Also, greedy approaches to save computational effort were designed.

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

Our greedy approach includes two algorithms, the first one based on conditioned 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 and SP problems. Based on our computational experiments we can expect for small problems:

1. if $k = 2$ the S-Q and Sr-Qr algorithms may be used with tolerable execution times and if $k = 3$ the same remark is also valid often,
2. if $k \leq 3$ the Q-Greedy, T-Greedy, Qr-Greedy and Tr-Greedy algorithms obtain optimal or near optimal solutions, and for $k \leq 6$ the algorithms obtain a good set of solutions with a tolerable computational effort,
3. for $k \leq 6$ the Q-Greedy and Qr-Greedy algorithms may be used to obtain a good set of solutions quickly.

In order to use the algorithms for medium to large problems it will be necessary to improve its performance (in the extensions presented in this section two possible improvements are mentioned).

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

1. 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 MSR(k) and MrelSR(k) problems.
2. 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:

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

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

3. 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 2014).
4. The algorithms could be improved if several constraints are added in order to break the symmetries (note that the output of the algorithms is a set and then there are several equivalent representations with our integer programming models).
5. We point out that our approach may be used in practice if the cost vectors become known over the time. Considering again our example in the introduction, that may be the case if the travel times and the demands of an emergency system are monitored in real time. In that scenario, we can choose a new solution when the system changes enough. Multiple changes in a short time, even by using a small set of solutions, may be a problem for humans and that should be considered for further works.
6. The approach used may be generalized easily (from the theoretically point of view at least) in order to consider other uncertainty models (finite uncertainty and bounded uncertainty).
7. The T-greedy (Tr-greedy) algorithm may be improved as follows: If $k = 2$ and beginning with $h^{(1)} \in X$ use the T-greedy (Tr-greedy) algorithm to find $H = \{h^{(1)}, h^{(2)}\}$. Next, beginning with $h^{(2)}$ use the T-greedy (Tr-greedy) algorithm to find $H = \{h^{(2)}, h^{(3)}\}$ and so on until convergence. If $k = 3$ similar approaches may be defined.

Appendix A: Proof of Proposition 1

(i) Let $\mathbf{c} \in \Omega$ and let $\mathbf{h}, \mathbf{x} \in X$.

- Let $I_{1,0} = \{i : \mathbf{h}_i = 1, \mathbf{x}_i = 0, 1 \leq i \leq n\}$
- Let $I_{0,1} = \{i : \mathbf{h}_i = 0, \mathbf{x}_i = 1, 1 \leq i \leq n\}$

then:

$$\mathbf{c}^t \mathbf{h} - \mathbf{c}^t \mathbf{x} = \sum_{i \in I_{1,0}} \mathbf{c}_i - \sum_{i \in I_{0,1}} \mathbf{c}_i \leq \sum_{i \in I_{1,0}} \mathbf{U}_i - \sum_{i \in I_{0,1}} \mathbf{L}_i = \mathbf{c}^+(\mathbf{x})^t \mathbf{h} - \mathbf{c}^+(\mathbf{x})^t \mathbf{x}$$

(ii) Let $\mathbf{x} \in X$ and let $\mathbf{c} \in \Omega$. Let \mathbf{h} be an optimal solution for $P(\mathbf{c}^+(\mathbf{x}))$.

- Let $I_{1,0} = \{i : \mathbf{h}_i = 1, \mathbf{x}_i = 0, 1 \leq i \leq n\}$
- Let $I_{0,1} = \{i : \mathbf{h}_i = 0, \mathbf{x}_i = 1, 1 \leq i \leq n\}$

then:

$$\begin{aligned} 0 &< \mathbf{c}^+(\mathbf{x})^t \mathbf{x} - v(P(\mathbf{c}^+(\mathbf{x}))) = \mathbf{c}^+(\mathbf{x})^t \mathbf{x} - \mathbf{c}^+(\mathbf{x})^t \mathbf{h} \\ &= \sum_{i \in I_{0,1}} \mathbf{L}_i - \sum_{i \in I_{1,0}} \mathbf{U}_i \leq \sum_{i \in I_{0,1}} \mathbf{c}_i - \sum_{i \in I_{1,0}} \mathbf{c}_i = \mathbf{c}^t \mathbf{x} - \mathbf{c}^t \mathbf{h} \end{aligned}$$

therefore: $v(P(\mathbf{c})) \leq \mathbf{c}^t \mathbf{h} < \mathbf{c}^t \mathbf{x}$.

- (iii) Let $\mathbf{x} \in X$ and let $\mathbf{c} \in \Omega$, then: $\mathbf{c}^+(\mathbf{x})^t \mathbf{x} \stackrel{(1)}{=} \mathbf{L}^t \mathbf{x} \stackrel{(2)}{\leq} \mathbf{c}^t \mathbf{x}$ where (1) From the definition of $\mathbf{c}^+(\mathbf{x})$ and (2) Since $\mathbf{L} \leq \mathbf{c}$ •.

Appendix B: Proof of Proposition 2

Let $X \subseteq \{0, 1\}^n$. Let $f : X \mapsto \mathbb{R}$ and $g : X \mapsto \mathbb{R}$ with $f(\mathbf{x}) > 0$ and $g(\mathbf{x}) > 0$ for all $\mathbf{x} \in X$. Let $P1$ be a problem in (\mathbf{x}) defined as:

$$(P1) : \max_{\mathbf{x} \in X} \frac{f(\mathbf{x})}{g(\mathbf{x})}$$

Let $\mu \geq 0$ and let $P2(\mu)$ be a problem in (\mathbf{x}) defined as;

$$(P2(\mu)) : \max_{\mathbf{x} \in X} f(\mathbf{x}) - \mu g(\mathbf{x})$$

then:

If $v(P2(\mu)) = 0$ then $f(\mathbf{x}) - \mu g(\mathbf{x}) \leq 0$ for all $\mathbf{x} \in X$, therefore $\frac{f(\mathbf{x})}{g(\mathbf{x})} \leq \mu$ for all $\mathbf{x} \in X$ and we have $v(P1) \leq \mu$. If $v(P2(\mu)) = 0$ and $v(P1) < \mu$ we have $\frac{f(\mathbf{x})}{g(\mathbf{x})} < \mu$ for all $\mathbf{x} \in X$ and then $f(\mathbf{x}) - \mu g(\mathbf{x}) < 0$ for all $\mathbf{x} \in X$ and we have a contradiction.

If $\mu = v(P1)$ and $v(P2(\mu)) > 0$ then we have $\frac{f(\mathbf{x})}{g(\mathbf{x})} > \mu = v(P1)$ for some $\mathbf{x} \in X$ and we have a contradiction, therefore $v(P2(\mu)) \leq 0$. If \mathbf{x}^* is an optimal solution for $P1$ then we have $f(\mathbf{x}^*) - \mu g(\mathbf{x}^*) = 0$ and then $v(P2(\mu)) \geq 0$ and finally $v(P2(\mu)) = 0$.

Hence we have:

- (1) $v(P2(\mu)) = 0$ if and only if $\mu = v(P1)$.
If $v(P2(\mu)) = 0$ and \mathbf{x}^* is an optimal solution for $P2(\mu)$ we have $v(P2(\mu)) = f(\mathbf{x}^*) - \mu g(\mathbf{x}^*) = f(\mathbf{x}^*) - v(P1)g(\mathbf{x}^*) = 0$ and then $v(P1) = \frac{f(\mathbf{x}^*)}{g(\mathbf{x}^*)}$, therefore \mathbf{x}^* is an optimal solution for $P1$. Hence we have:
- (2) If $v(P2(\mu)) = 0$ and \mathbf{x}^* is an optimal solution for $P2(\mu)$ then \mathbf{x}^* is an optimal solution for $P1$.

Let $X = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(L)}\}$ then

$$v(P2(\mu)) = \max\{f(\mathbf{x}^{(1)}) - \mu g(\mathbf{x}^{(1)}), \dots, f(\mathbf{x}^{(L)}) - \mu g(\mathbf{x}^{(L)})\}.$$

Since $g(\mathbf{x}) > 0$ for all $\mathbf{x} \in X$ then we have:

- (3) $v(P2(\mu))$ is a piecewise linear and decreasing function in μ .

We have:

$$v(QX_r(H)) = \max_{\mathbf{x} \in X} \frac{\min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{x})^t \mathbf{h} - \mathbf{c}^+(\mathbf{x})^t \mathbf{x}}{\mathbf{c}^+(\mathbf{x})^t \mathbf{x}} = v(RX_r(H)) - 1$$

Let $f(\mathbf{x}) = \min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{x})^t \mathbf{h}$ and let $g(\mathbf{x}) = \mathbf{c}^+(\mathbf{x})^t \mathbf{x}$. We use (1),(2) and (3) to find:

- (i) $v(RX_r(\mu, H)) = 0$ if and only if $\mu = v(QX_r(H)) + 1$.
- (ii) If $v(RX_r(\mu, H)) = 0$ and \mathbf{x}^* is an optimal solution for $RX_r(\mu, H)$ then \mathbf{x}^* is an optimal solution for $QX_r(H)$.
- (iii) $v(RX_r(\mu, H))$ is a piecewise linear and decreasing convex function in μ •

Appendix C: Proof of Proposition 3

Let $X = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(L)}\}$, let $\phi_j = \min_{\mathbf{h} \in H} \mathbf{c}^+(\mathbf{x}^{(j)})^t \mathbf{h}$ and let $\delta_j = \mathbf{c}^+(\mathbf{x}^{(j)})^t \mathbf{x}^{(j)}$ for $j = 1, \dots, L$.

The algorithm may be rewritten as follows:

Let $\mu_1 = 1$.

Algorithm Find- μ^*

1. $i = 1$.
2. Solve $R_r(\mu_i, H)$. Let $\mathbf{x}^{(j_i)}$ be an optimal solution.
3. If $v(R_r(\mu_i, H)) = 0$ let $\mu^* = \mu_i$, let $\mathbf{x}^* = \mathbf{x}^{(j_i)}$ and stop.
4. Let $\mu_{i+1} = \frac{\phi_{j_i}}{\delta_{j_i}}$, let $i = i + 1$ and return to Step 2.

If $v(R_r(\mu_1, H)) > 0$ then $\mu_2 = \frac{\phi_{j_1}}{\delta_{j_1}}$ and $\phi_{j_1} - \mu_1 \delta_{j_1} > 0$. Hence $\mu_2 > 1 = \mu_1$.

If $v(R_r(\mu_2, H)) > 0$ then $\mu_3 = \frac{\phi_{j_2}}{\delta_{j_2}}$ and $\phi_{j_2} - \mu_2 \delta_{j_2} > 0$. Hence $\mu_3 > \mu_2$.

In general we have that $\mu_i < \mu_{i+1}$ for all i . Since X is a finite set then the sequence $\mu_1, \dots, \mu_s, \dots$ generated by the algorithm must be finite and then $v(R_r(\mu_i, H)) = 0$ for some i (otherwise the algorithm generates an infinite sequence of μ -values).

Appendix D

See Tables 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 and 15.

Table 4 SP problems with mesh topology. Q-greedy, T-greedy and S-Q algorithms for the $MSR(k)$ problem. Times, failures and iterations. 10 problems in each set

γ	r	m	k	tQ	itT	tT	f	itS-Q	tS-Q	avetQ	avetT	avetS-Q	
0.15	10	25	2	0.22	7	2.53	0	10	4.80	0.18	1.65	4.12	
			3	0.56	10	4.51	0	16	14.06	0.37	2.89	8.75	
		50	2	0.63	16	14.51	0	30	71.81	0.40	6.21	26.47	
			3	1.32	19	18.16	0	39	182.26	0.92	8.99	53.30	
		75	2	1.39	14	27.74	0	27	166.67	0.96	12.19	61.24	
			3	2.45	18	37.54	1	30	242.91	1.77	17.91	126.78	
	15	25	2	0.40	9	6.77	0	16	25.32	0.28	2.60	8.03	
			3	0.82	12	10.10	0	21	58.15	0.52	3.95	13.44	
		50	2	1.22	8	13.86	0	17	72.01	0.88	8.84	34.58	
			3	2.28	11	20.64	0	21	153.56	1.68	13.35	70.17	
		75	2	2.10	13	34.33	0	27	288.96	1.57	17.87	90.00	
			3	3.92	21	64.84	1	25	1081.79	3.25	29.42	310.82	
0.35	10	25	2	0.60	9	3.97	0	17	15.70	0.25	2.37	7.68	
			3	0.61	15	7.99	0	29	69.83	0.43	4.19	22.38	
		50	2	1.01	26	30.66	0	55	3630.66	0.56	14.56	542.39	
			3	2.05	33	41.23	4	40	657.31	1.05	22.49	256.62	
		15	25	2	0.84	7	6.85	0	11	22.82	0.57	4.59	15.03
				3	1.53	10	11.03	0	18	101.71	1.15	7.51	41.34
	50	2	2	2.01	24	50.45	0	44	334.27	1.15	24.93	159.83	
			3	3.82	29	62.34	3	33	650.07	2.10	35.48	341.38	

Table 5 SP problems with mesh topology. Q-greedy, T-greedy and S-Q algorithms for the $MSR(k)$ problem. Reduction percentage for the regret values. 10 problems in each set

γ	r	m	redQ2	redT2	redS-Q2	dif2	redQ3	redT3	redS-Q3	dif3	
0.15	10	25	26.95	63.03	63.03	0.00	60.53	76.68	76.68	0.00	
		50	33.00	58.17	58.17	0.00	54.06	64.83	66.67	1.84	
		75	12.07	44.55	44.97	0.42	32.56	52.10	55.29	3.18	
	15	25	15.66	55.30	56.61	1.32	66.22	75.31	79.63	4.32	
		50	24.34	52.94	52.94	0.00	54.28	56.76	62.51	5.75	
		75	15.26	35.85	37.13	1.28	35.43	49.57	50.66	1.09	
0.35	10	25	14.42	25.25	26.04	0.79	37.60	44.93	50.08	5.16	
		50	12.67	19.85	22.84	2.99	21.20	28.18	37.32	9.14	
		15	25	13.58	41.17	42.53	1.36	46.48	53.37	55.55	2.18
	50		15.50	27.49	34.92	7.42	25.06	39.63	46.85	7.22	
	AVE						1.52				

Table 6 SP problems with euclidean topology. Q-greedy, T-greedy and S-Q algorithms for the $MSR(k)$ problem. Times and iterations. 10 problems in each set

p	$ V $	k	tQ	itT	tT	f	it-SQ	tS-Q	avetQ	avetT	avetS-Q
0.35	15	2	0.15	10	2.35	0	20	6.68	0.13	1.04	2.84
		3	0.24	14	3.10	0	28	12.27	0.21	1.49	4.79
	20	2	0.40	11	14.96	0	28	19.01	0.21	3.42	8.79
		3	0.55	14	15.45	1	33	1352.98	0.34	4.16	162.40
	25	2	0.53	13	6.76	0	26	27.17	0.29	2.61	9.07
		3	0.72	16	9.96	0	36	602.60	0.42	3.65	219.22
30	2	0.57	14	5.64	0	32	30.76	0.39	3.57	15.90	
	3	0.84	19	11.40	3	32	3603.13	0.55	5.21	587.11	
0.70	15	2	0.31	15	3.68	0	31	10.30	0.15	1.41	3.21
		3	0.37	19	4.57	0	49	54.19	0.22	1.87	9.03
	20	2	0.35	20	5.39	0	33	12.53	0.26	2.33	6.66
		3	0.43	26	6.88	0	49	69.11	0.34	3.10	18.59
	25	2	0.64	20	11.53	0	52	183.90	0.31	3.40	39.35
		3	0.81	24	13.28	1	57	516.06	0.42	4.29	72.90
30	2	0.57	19	8.02	2	27	19.80	0.34	3.58	11.58	
	3	0.82	29	10.53	3	43	3603.65	0.48	4.94	698.33	

Table 7 SP problems with euclidean topology. Q-greedy, T-greedy and S-Q algorithms for the $MSR(k)$ problem . Reduction percentage for the regret values. 10 problems in each set

p	$ V $	redQ2	redT2	redS-Q2	dif2	redQ3	redT3	redS-Q3	dif3
0.35	15	17.20	17.76	18.18	0.42	22.50	24.16	26.68	2.52
	20	2.38	3.90	5.43	1.53	10.16	11.37	14.04	2.67
	25	3.42	4.40	5.42	1.02	8.90	10.20	10.47	0.27
	30	2.47	2.79	3.65	0.86	6.51	6.51	11.01	4.50
0.70	15	23.49	32.83	35.29	2.45	34.21	41.35	44.53	3.18
	20	17.33	24.20	26.23	2.03	26.61	33.02	35.65	2.63
	25	15.20	17.69	19.25	1.56	22.86	25.19	26.53	1.35
	30	11.42	12.62	13.83	1.21	19.47	19.14	22.53	3.39
AVE					1.39				2.56

Table 8 p -M problems. Q-greedy, T-greedy and S-Q algorithms for the $MSR(k)$ problem. Times, failures and iterations. 10 problems in each set

γ	m	p	k	tQ	itT	tT	f	itS-Q	tSQ	avetQ	avetT	avetS-Q
0.15	30	2	2	0.61	9	63.93	0	16	66.76	0.50	9.50	15.60
			3	0.92	12	65.04	0	19	70.72	0.80	11.06	18.92
		5	2	0.7	10	5.54	0	20	20.93	0.49	3.34	10.53
	3		1.17	13	9.47	0	22	64.59	0.79	5.28	18.82	
	60	2	2	1.96	9	19.48	0	18	248.98	1.44	14.15	118.45
			3	3.67	13	26.31	0	28	751.31	2.52	19.92	344.83
		5	2	2.00	17	30.33	0	40	3630.33	1.24	23.22	563.49
	3		3.68	20	37.67	3	45	3260.77	2.14	30.70	1280.83	
	80	2	2	2.26	9	31.12	0	18	246.37	1.93	21.86	173.92
			3	4.61	13	51.19	3	25	1448.88	3.64	36.35	917.23
		5	2	3.25	20	65.40	0	41	1439.35	2.54	41.11	430.85
	3		5.77	24	84.88	5	35	2620.62	3.77	58.66	1573.66	
0.35	30	2	2	0.62	7	4.82	0	15	19.54	0.47	3.15	11.82
			3	0.97	10	7.63	0	18	48.93	0.85	4.96	25.74
		5	2	0.64	16	13.42	0	30	48.33	0.53	6.15	25.42
	3		1.18	21	19.37	1	45	762.46	0.86	8.72	124.84	
	60	2	2	2.19	10	24.30	0	19	197.65	1.41	16.71	108.08
			3	3.42	14	35.90	2	28	1154.23	2.46	24.75	643.08
		5	2	1.52	18	36.49	1	39	1891.05	1.17	24.50	546.48
	3		3.16	26	61.84	10	26	–	2.40	38.05	–	
	80	2	2	2.66	10	41.96	0	20	297.20	2.27	29.83	201.44
			3	4.28	14	63.50	6	23	2960.86	3.71	47.28	1869.17

Table 9 p -M problems. Q-greedy, T-greedy and S-Q algorithms for the $MSR(k)$ problem. Reduction percentage for the regret values. 10 problems in each set

γ	m	p	redQ2	redT2	redS-Q2	dif2	redQ3	redT3	redS-Q3	dif3
0.15	30	2	24.25	27.32	27.37	0.05	33.03	38.28	45.83	7.56
		5	18.93	33.45	37.30	3.85	40.02	46.91	51.35	4.44
	60	2	7.63	10.63	13.12	2.49	17.37	17.35	24.54	7.20
		5	12.92	24.16	26.24	2.08	23.76	30.10	37.28	7.18
	80	2	8.80	9.22	9.67	0.45	16.71	18.27	21.31	3.04
5		15.89	21.81	22.93	1.11	22.43	26.27	33.52	7.25	
0.35	30	2	7.93	9.69	9.78	0.09	21.79	24.38	26.81	2.44
		5	9.92	22.00	22.83	0.83	20.91	29.81	37.36	7.55
	60	2	6.71	7.96	8.33	0.37	10.67	12.73	13.48	0.75
		5	12.49	19.29	20.03	0.74	19.42	23.29	32.71	9.43
80	2	3.33	4.75	5.27	0.52	7.70	8.85	18.05	9.20	
AVE						1.14				6.00

Table 10 SP problems with mesh topology. Qr-greedy, Tr-greedy and Sr-Qr algorithms for the *MrelSR(k)* problem. Times, failures and iterations. 10 problems in each set

γ	r	m	k	tQr	itTr	tTr	f	itSr-Qr	tSr-Qr	avetQr	avetTr	avetSr-Qr	
0.35	10	25	2	2.55	8	4.21	0	16	17.77	1.29	2.58	6.47	
			3	3.17	10	6.65	3	19	32.80	1.83	4.37	16.89	
		50	2	11.24	22	48.22	0	55	2607.07	7.33	19.56	411.9	
			3	12.21	26	59.27	3	28	178.08	8.78	29.76	126.20	
		15	25	2	6.90	8	12.10	0	21	82.83	4.39	7.45	26.24
				3	10.06	12	22.72	0	37	796.68	5.72	13.02	119.1
	50		2	26.06	18	51.81	0	47	1117.10	10.60	23.48	200.28	
			3	27.85	25	88.11	2	34	3332.92	12.63	42.58	824.52	

Table 11 SP problems with mesh topology. Qr-greedy, Tr-greedy and Sr-Qr algorithms for the *MrelSR(k)* problem . Reduction percentage for the regret values. 10 problems in each set

γ	r	m	redQr2	redTr2	redSr-Qr2	difr2	redQr3	redTr3	redSr-Qr3	difr3
0.35	10	25	44.66	52.60	53.85	1.26	65.80	62.15	71.03	8.88
		50	11.27	29.49	31.11	1.61	18.01	38.92	43.72	4.80
	15	25	27.24	31.30	32.83	1.52	42.17	46.57	53.33	6.76
		50	14.29	23.90	24.04	0.14	20.23	33.33	37.50	4.17
AVE						1.13				5.15

Table 12 SP problems with euclidean topology. Qr-greedy, Tr-greedy and Sr-Qr algorithms for the *MrelSR(k)* problem. Times, failures and iterations. 10 problems in each set

p	$ V $	k	tQr	itTr	tTr	f	itSr-Qr	tSr-Qr	avetQr	avetTr	avetSr-Qr	
0.35	15	2	1.17	8	2.48	0	15	7.80	0.90	1.68	4.54	
			3	1.33	12	3.79	0	24	12.17	1.08	2.64	6.80
		20	2	1.60	8	2.72	0	21	14.55	1.02	1.91	6.52
			3	2.03	12	4.99	0	28	25.25	1.29	3.38	15.34
		25	2	2.57	12	7.11	0	27	28.37	1.67	3.18	10.66
			3	2.85	16	9.69	0	37	1570.36	1.95	5.11	173.67
	30	2	5.63	10	10.95	0	23	41.39	2.84	5.40	22.52	
		3	6.25	16	19.60	4	30	116.25	3.24	9.02	84.78	
	0.70	15	2	2.46	14	4.49	0	26	11.93	1.02	1.80	4.80
			3	2.56	17	5.33	0	28	14.06	1.13	2.54	6.93
		20	2	3.64	17	6.39	0	35	22.70	1.66	2.73	8.91
			3	3.86	23	8.78	0	58	1501.12	1.83	3.96	167.78
25		2	5.14	17	8.78	1	35	32.00	2.00	3.76	13.36	
		3	5.50	23	11.81	1	52	670.81	2.24	5.84	94.17	
30	2	6.04	19	7.32	0	34	27.46	3.12	4.41	14.79		
	3	6.28	23	9.53	2	45	1163.13	3.42	6.29	214.86		

Table 13 SP problems with euclidean topology. Qr-greedy, Tr-greedy and Sr-Qr algorithms for the $MrelSR(k)$ problem. Reduction percentage for the regret values. 10 problems in each set

p	$ V $	redQr2	redTr2	redSr-Qr2	difr2	redQr3	redTr3	redSr-Qr3	dif3
0.35	15	10.15	10.69	12.38	1.69	19.48	19.78	21.59	1.81
	20	3.94	7.90	9.12	1.21	16.33	17.19	18.34	1.16
	25	6.46	7.21	7.72	0.52	11.72	13.59	14.78	1.18
	30	3.63	3.69	4.03	0.34	7.58	8.48	11.10	2.62
0.70	15	16.53	23.74	26.62	2.87	26.59	35.04	39.83	4.79
	20	18.34	30.57	30.72	0.15	32.68	37.81	38.65	0.84
	25	12.69	14.10	15.10	0.99	20.05	20.81	25.85	5.04
	30	15.09	16.71	18.32	1.61	19.11	22.18	28.06	5.88
AVE					1.17				2.91

Table 14 p -M problems. Qr-greedy, Tr-greedy and Sr-Qr algorithms for the $MrelSR(k)$ problem. Times, failures and iterations. 10 problems in each set

γ	m	p	k	tQr	itTr	tTr	f	itSr-Qr	tSr-Qr	avetQr	avetTr	avetSr-Qr
0.15	30	2	2	3.34	7	5.16	0	10	8.86	2.52	3.76	7.79
			3	3.90	9	7.81	0	16	28.11	3.20	6.69	14.67
		5	2	3.75	10	8.70	0	16	25.02	2.55	6.00	13.17
			3	4.31	12	12.10	0	23	50.51	3.33	9.36	22.30
	60	2	2	14.64	9	30.30	0	18	145.30	11.99	22.06	81.95
			3	16.91	11	40.11	0	21	371.19	14.33	34.53	185.96
		5	2	30.60	13	47.90	0	31	351.97	19.17	32.41	148.64
			3	33.17	16	64.28	0	36	3214.08	21.54	45.98	927.00
	80	2	2	32.17	9	48.18	0	19	240.94	20.16	34.44	150.71
			3	35.31	11	65.37	0	24	2502.77	23.65	53.08	612.01
		5	2	74.31	19	163.79	0	40	1973.99	37.56	74.60	833.89
			3	79.90	22	189.13	5	32	3690.50	40.85	106.01	1433.96

Table 15 p -M problems. Qr-greedy, Tr-greedy and Sr-Qr algorithms for the $MrelSR(k)$ problem. Reduction percentage for the regret values. 10 problems in each set

γ	m	p	redQr2	redTr2	redSr-Qr2	difr2	redQr3	redTr3	redSr-Qr3	dif3
0.15	30	2	26.84	30.30	31.62	1.32	49.34	50.37	51.71	1.34
		5	15.38	32.51	33.34	0.83	44.74	53.22	54.48	1.26
	60	2	16.67	18.71	18.71	0.00	27.23	27.61	30.49	2.88
		5	13.64	21.68	23.48	1.81	21.93	30.30	34.66	4.36
	80	2	13.79	14.65	16.81	2.16	23.19	24.01	27.20	3.20
		5	7.82	15.33	18.33	3.00	13.48	22.90	31.48	8.58
AVE						1.52				3.60

References

- 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
- Averbakh I (2005) Computing and minimizing the relative regret in combinatorial optimization with interval data. *Discrete Optim* 2:273–287
- Boffey TB, Karkazis J (1984) p-Medians and multi-medians. *J Oper Res Soc* 35(1):57–64
- Buchheim C, Kurtz J (2016) Min–max–min robustness: a new approach to combinatorial optimization under uncertainty based on multiple solutions. *Electron Not Discrete Math* 52:45–52
- Buchheim C, Kurtz J (2017) Min–max–min robust combinatorial optimization. *Math Program* 163(1–2):1–23
- Chassein A, Goerigk M, Kurtz J, Poss M (2019) Faster algorithms for min–max–min robustness for combinatorial problems with budgeted uncertainty. *Eur J Oper Res* 279(2):308–319
- Candia-Véjar A, Alvarez-Miranda E, Maculan N (2011) Minmax regret combinatorial optimization problems: an algorithmic perspective. *RAIRO-Oper Res* 45:101–129
- Carlsson C, Fuller R (2012) Fuzzy reasoning in decision making and optimization. *Physica*, vol 82. Springer, Berlin
- Chassein A, Goerigk M (2016) Performance analysis in robust optimization. In: Doumpos M, Zopounidis C, Grigoroudis E (eds) *Robustness analysis in decision aiding, optimization, and analytics*. International series in operations research and management science, vol 241. Springer, Cham
- Crema A (2000) An algorithm for the multiparametric 0–1-integer linear programming problem relative to the objective function. *Eur J Oper Res* 125:18–24
- Crema A (2014) Mathematical programming approach to tighten a Big- M formulation. www.optimization-online.org. Accessed Aug 2014
- Hanasusanto G, Kuhn D, Wiesemann W (2015) K-adaptability in two-stage robust binary programming. *Oper Res* 63(4):877–891
- Kasperski A, Zielinski P (2016) Robust discrete optimization under discrete and interval uncertainty: a survey. In: Doumpos M, Zopounidis C, Grigoroudis E (eds) *Robustness analysis in decision aiding, optimization, and analytics*. International series in operations research and management science. Springer, Berlin
- Kouvelis P, Yu G (2013) *Robust discrete optimization and its applications*, vol 14. Springer, Berlin
- Li J, Liu Y (2016) Approximation algorithms for stochastic combinatorial optimization problems. *J Oper Res Soc China* 4(1):1–47
- Megiddo N (1979) Combinatorial optimization with fractional objective functions. *Math Oper Res* 4(4):414–424
- Montemanni R, Gambardella LM (2005) The robust shortest path problem with interval data via Benders decomposition. *4 OR* 3:315–328
- Oberdieck R, Diangelakis NA, Nascu I, Papatathanasiou MM, Sun M, Avraamidou S, Pistikopoulos EN (2016) On multi-parametric programming and its applications in process systems engineering. *Chem Eng Res Des* 116:61–82

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.