

Absolute regret of implicitly defined sets for combinatorial optimization problems

Alejandro Crema

*Escuela de Computación, Facultad de Ciencias, Universidad Central de Venezuela, Caracas
1041, Venezuela.*

email: alejandrocrema@gmail.com, alejandro.crema@ciens.ucv.ve

Abstract

We consider combinatorial optimization problems with interval uncertainty in the cost vector. Recently a new approach was developed to deal with such uncertainties: instead of a single one absolute robust solution, obtained by solving a min max problem, a set of cardinality predefined and minimal absolute regret, obtained by solving a min max min problem, is considered. With that approach, the set of solutions is computed once and then 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 that approach by presenting an algorithm to compute the absolute regret of an implicitly defined set and a greedy heuristic to deal with the min max min absolute regret problem for a practical interest case.

Keywords: Robustness and sensitivity analysis, Combinatorial optimization, Mixed integer programming, Min max min absolute regret.

1. Introduction

Let $X \subseteq \{0, 1\}^n$ with $X \neq \emptyset$. Let $c \in \mathbb{R}^n$, let $Y \subseteq X$ with $Y \neq \emptyset$ and let $P(c, Y)$ be a combinatorial optimization problem in (x) defined as follows:

$$P(c, Y) : \min_{x \in Y} c^t x$$

In this paper Y admits a 0-1-Mixed Integer Linear Programming (0-1-MILP) formulation for all $Y \subseteq X$ considered.

1.1. Interval uncertainty for the cost vector

Data uncertainty appears in many optimization problems. There are several options to model the uncertainty of the cost vector of a combinatorial optimization problem. Some examples are $([1], [2])$: finite, convex, interval, bounded, ellipsoidal and polytopal uncertainty.

We consider the Interval Uncertainty for the cost vector as follows:

Let $l, u \in \mathbb{R}^n$ with $0 \leq l \leq u$, and let $\Omega = \{c \in \mathbb{R}^n : l \leq c \leq u\}$.

1.2. Absolute regret of sets

In recent decades Robust ([1], [2]), Stochastic ([3]), Multiparametric [4]) and Fuzzy programming ([5]) 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 ([1]): classic robustness, absolute or relative regret robustness, adjustable robustness, recoverable robustness, light robustness, soft robustness, lexicographic α -robustness, recovery-to-optimality, or similarity-based robustness. In this paper we consider the absolute regret robustness.

In this paper if S is an optimization problem then $v(S)$ is its optimal value and $F(S)$ is its feasible solutions set. If S is a minimization problem and $F(S) = \emptyset$ then $v(S) = +\infty$

Let $Y \subseteq X$ with $Y \neq \emptyset$. Let $Q(Y)$ be a problem in (c) defined as follows:

$$Q(Y) : \max_{c \in \Omega} \{v(P(c, Y)) - v(P(c, X))\}$$

The absolute regret of Y is defined as follows: $R(Y) = v(Q(Y))$. Note that if $Y^{(1)} \subseteq Y^{(2)} \subseteq X$ with $Y^{(1)} \neq \emptyset$ then $0 = R(X) \leq R(Y^{(2)}) \leq R(Y^{(1)}) < \infty$.

Let $x \in X$. If $Y = \{x\}$ then $R(Y) = R(\{x\})$ is the usual absolute regret of x . From now $R(\{x\})$ will be rewritten as $R(x)$.

Let $x^{(1)}, \dots, x^{(k)} \in X$. If Y is defined explicitly as $Y = \{x^{(1)}, \dots, x^{(k)}\}$ then $R(Y)$ was studied in [6] inspired on several previous works that use classic robustness and convex uncertainty ([7], [8]).

In this paper we study the absolute regret of implicitly defined sets which satisfies some practical interest property.

1.3. Min max min absolute regret problem

Let Δ be a non-empty family of subsets of X . Since X is a finite set then Δ is finite. We want to find a set Y from Δ with minimal absolute regret.

The min max min absolute regret problem based on Δ ($MR(\Delta)$) is a problem in (Y) defined as follows:

$$MR(\Delta) : \min_{Y \in \Delta} R(Y) = \min_{Y \in \Delta} v(Q(Y)) = \min_{Y \in \Delta} \max_{c \in \Omega} \{v(P(c, Y)) - v(P(c, X))\} =$$

$$\min_{Y \in \Delta} \max_{c \in \Omega} \left\{ \min_{x \in Y} c^t x - \min_{x \in X} c^t x \right\}$$

If $\Delta = \{Y \subseteq X : Y = \{x\}, x \in X\}$ then $MR(\Delta)$ is the usual min max problem to find an absolute robust solution as follows:

$$\begin{aligned} MR(\Delta) : \min_{Y \in \Delta} R(Y) &= \min_{Y \in \Delta} v(Q(Y)) = \min_{x \in X} v(Q(\{x\})) = \\ &= \min_{x \in X} \max_{c \in \Omega} \{c^t x - v(P(c, X))\} = R(x_R) \end{aligned}$$

with x_R an absolute robust solution.

Let $k \geq 1$. Let $\Delta(k) = \{Y \subseteq X : |Y| = k\}$ then $MR(\Delta(k))$ was considered and studied in [6]. The min max min approach for $\Delta(k)$ has been considered with classical robustness and convex uncertainty ([7], [8]), [9]).

Let Y be an optimal solution for $MR(\Delta)$. With this new approach Y may be computed once and then we can solve $P(c, Y)$ instead of $P(c, X)$ in order to use a set of solutions that satisfies the property of practical interest that is used to define Δ .

The framework considered in this paper for $MR(\Delta)$ is defined as follows: let $r \in [n]$, let $\Phi \subseteq \{0, 1\}^r$ and if $\phi \in \Phi$ let $X(\phi) \subseteq X$ with $X(\phi) \neq \emptyset$. The sets $X(\phi) \forall \phi \in \Phi$ and the set Φ admit a 0-1-MILP formulation.

The problem is:

$$MR(\Delta) : \min_{Y \in \Delta} R(Y) = \min_{\phi \in \Phi} R(X(\phi))$$

1.4. Paper organization

In section 2 we present an algorithm to solve $Q(Y)$. In section 3 we present a Δ -case of practical interest, called Δ_k . In section 4 we present a greedy heuristic to deal with $MR(\Delta_k)$. Computational experience may be seen in section 5. Conclusions may be seen in section 6. For the sake of completeness we present in Appendix A an optimal algorithm to solve $MR(\Delta)$ for the framework presented with its drawbacks which justify the use of heuristics.

Several results presented in [6] to $\Delta(k)$ are valid to a general Δ with the proofs valid without changes. If that is the case the result, rewritten for the new context, will be presented without a proof.

All algorithms that we present in this paper may be executed using a commercial software, for example ILOG-Cplex 12.10 (<http://ibm-ilog-cplex-optimization-studio-acade.software.informer.com/12.10/>) when all problems to be solved during its execution admit a 0-1-Mixed Integer Linear Programming (0-1-MILP) formulation. The 0-1-MILP formulation to each problem considered is presented opportunely.

1.5. Main contribution

To the best of our knowledge algorithms to find the absolute regret and to solve the min max min absolute regret problem for implicitly defined sets have not been studied so far.

2. An algorithm to find the absolute regret of implicitly defined sets

Next we present an algorithm to solve $Q(Y)$.

2.1. Theoretical results

The next lemma will be use to rewrite $Q(Y)$.

Lemma 1. *Let $Y \subseteq X$ with $Y \neq \emptyset$. Let $c^+(x)_j = l_j x_j + u_j(1 - x_j) \quad \forall j \in [n], \forall x \in X$. Let $Q1(Y)$ be a problem in (x) defined as follows:*

$$Q1(Y) : \max_{x \in X} \{ \min_{y \in Y} c^+(x)^t y - l^t x \}$$

then:

(1) *Let x be an optimal solution for $Q1(Y)$ then $c^+(x)$ is an optimal solution for $Q(Y)$, x is an optimal solution for $P(c^+(x), X)$ and $v(P(c^+(x), X)) = c^+(x)^t x = l^t x$.*

(2) *Let c be an optimal solution for $Q(Y)$ and let x be an optimal solution for $P(c, X)$ then x is an optimal solution for $Q1(Y)$.*

(3) $R(Y) = v(Q(Y)) = v(Q1(Y)) < \infty$.

Proof See lemmas 1 and 2 in [6] •

If Y is an explicitly defined set as follows: $Y = \{x^{(1)}, \dots, x^{(r)}\} \subseteq X$ then the next lemma tell us how to solve $Q1(Y)$.

Lemma 2. *Let $r \geq 1$. Let $Y = \{x^{(1)}, \dots, x^{(r)}\} \subseteq X$.*

Let $Q2(Y)$ be a 0-1-MILP problem in (σ, x) defined as follows:

$$\begin{aligned} Q2(Y) : \max \quad & \sigma - l^t x \quad s.t. \\ & \sigma \leq c^+(x)^t x^{(i)} \quad \forall i \in [r] \\ & \sigma \in \mathbb{R}, \quad x \in X \end{aligned}$$

then:

(1) *If x is an optimal solution for $Q1(Y)$ then (σ, x) is an optimal solution for $Q2(Y)$ with $\sigma = \min_{i \in [r]} c^+(x)^t x^{(i)}$ and x is an optimal solution for $P(c^+(x), X)$*

(2) If (σ, x) is an optimal solution for $Q2(Y)$ then x is an optimal solution for $Q1(Y)$ and for $P(c^+(x), X)$

$$(3) v(Q1(Y)) = v(Q2(Y))$$

Proof See lemmas 1 and 2 in [6] •

Let $Y \subseteq X$ implicitly defined with $Y \neq \emptyset$. We have problems to find upper and lower bounds of $v(Q1(Y))$, as follows:

$$\text{Let } x \in X \text{ then } v(P(c^+(x), Y)) - l^t x \leq v(Q1(Y))$$

Let $r \geq 1$ and let $W = \{x^{(1)}, \dots, x^{(r)}\} \subseteq Y$. Since $W \subseteq Y$ we have that: $v(Q2(W)) = v(Q1(W)) \geq v(Q1(Y))$

2.2. Algorithm

Next we present an algorithm to solve $Q1(Y)$ that iterates between $Q2(W)$ and $P(c^+(x), Y)$ for a sequence of W and x until the gap is closed.

Q2-P-Algorithm (Q2-P-A) to solve $Q1(Y)$:

Let $Y \subseteq X$ with $Y \neq \emptyset$, find $x \in Y$ and let $W = \{x\}$.

1. Solve $Q2(W)$ and let (σ^*, x^*) be an optimal solution.
2. Solve $P(c^+(x^*), Y)$ and let y^* be an optimal solution.
3. If $y^* \in W$ stop.
4. Let $W = W \cup \{y^*\}$ and return to step 1.

Lemma 3. *Q2-P-A stops in a finite number of iterations with $x^* \in X$ and $W \subseteq Y$ such that:*

$$v(Q2(W)) = v(Q1(Y)) = v(P(c^+(x^*), Y)) - l^t x^*$$

Proof: Let (σ^*, x^*) be an optimal solution for $Q2(W)$ (see step 1), let y^* be an optimal solution for $P(c^+(x^*), Y)$ (see step 2) and let us suppose that $y^* \in W$ (see step 3) then:

$$v(Q1(Y)) \stackrel{(1)}{\leq} v(Q2(W)) \stackrel{(2)}{=} \sigma^* - l^t x^* \stackrel{(3)}{=} \min_{x \in W} c^+(x^*)^t x - l^t x^* \stackrel{(4)}{\leq}$$

$$c^+(x^*)^t y^* - l^t x^* \stackrel{(5)}{=} v(P(c^+(x^*), Y)) - l^t x^* \stackrel{(6)}{\leq} v(Q1(Y))$$

with:

- (1) Since $v(Q2(W))$ is an upper bound

- (2) Since (σ^*, x^*) is an optimal solution for $Q2(W)$
- (3) From lemma 3
- (4) Since $y^* \in W$
- (5) Since y^* is an optimal solution for $P(c^+(x^*), Y)$
- (6) Since $v(P(c^+(x^*), Y)) - l^t x^*$ is a lower bound

Since Y is a finite set we have that $Q2$ - P -A stops in a finite number of iterations •

Note that when the algorithm stops we have $v(Q2(W)) = v(Q1(W)) = v(Q1(Y))$, however if $W \subset Y$ we have that:

$$\min_{x \in Y} c^t x \leq \min_{x \in W} c^t x$$

Note that $Q2(W)$ and $P(c^+(x^*), Y)$ are 0-1-MILP problems $\forall W \subseteq X, \forall Y \subseteq X$ and then the algorithm works for all Y that admits a 0-1-MILP formulation.

3. Δ -case of practical interest

In this paper if $q \in \mathbb{Z}$ with $q \geq 1$ let $[q] = \{1, \dots, q\}$ and if $x \in \{0, 1\}^n$ let $p_r(x)$ be the projection of x onto $\{0, 1\}^r$ defined as follows: $p_r(x)_i = x_i \ \forall i \in [r] \ \forall r \in [n]$.

3.1. Δ_k -case based on bounded projections

Next we present a Δ_k -case of practical interest based on bounded projections and illustrate its use with a emergency system based on a p -medians location with K coverage requirement and q locations $((q, p, K)$ -m) problem with ([10]). In Appendix A we present another Δ -case of practical interest to be studied in a future paper.

Let $r \in [n]$. Let x_R an absolute robust solution. Let $k \in [\|p_r(x_R)\|_1, r]$, let $X(\phi) = \{x : x \in X, p_r(x) \leq \phi\} \ \forall \phi \in \{0, 1\}^r$, let $\Phi_k = \{\phi \in \{0, 1\}^r : \|\phi\|_1 = k, X(\phi) \neq \emptyset\}$, and let $\Delta_k = \{Y : Y = X(\phi), \ \phi \in \Phi_k\}$.

Lemma 4. *Let $\phi^{(k)}$ be an optimal solution for $MR(\Delta_k)$ then:*

- (i) $R(X(\phi^{(k)})) \leq R(x_R)$
- (ii) $R(X(\phi^{(k+1)})) \leq R(X(\phi^{(k)}))$
- (iii) $R(\phi^{(r)}) = R(X) = 0$

Proof:

(i) Let $\phi \in \{0, 1\}^r$ such that $\|\phi\|_1 = k$ and $p_r(x_R) \leq \phi$ then: $x_R \in X(\phi)$, $\phi \in \Phi_k$ and $X(\phi) \in \Delta_k$. Since $\phi^{(k)}$ is an optimal solution for $MR(\Delta_k)$, $X(\phi) \in \Delta_k$ and $x_R \in X(\phi)$ we have that:

$$R(X(\phi^{(k)})) = v(MR(\Delta_k)) \leq R(X(\phi)) \leq R(x_R)$$

(ii) Let $\phi \in \{0, 1\}^r$ such that $\|\phi\|_1 = k+1$ and $\phi^{(k)} \leq \phi$ then $X(\phi^{(k)}) \subseteq X(\phi)$ and $\phi \in \Phi_{k+1}$. Since $\phi^{(k+1)}$ is an optimal solution for $MR(\Delta_{k+1})$, $X(\phi) \in \Delta_{k+1}$ and $X(\phi^{(k)}) \subseteq X(\phi)$ we have that:

$$R(X(\phi^{(k+1)})) = v(MR(\Delta_{k+1})) \leq R(X(\phi)) \leq R(X(\phi^{(k)}))$$

(iii) $\Phi_r = \{\phi \in \{0, 1\}^r : \|\phi\|_1 = r, X(\phi) \neq \emptyset\}$ then we have that $\Phi_r = \{\phi\}$ with $\phi_i = 1 \ \forall i \in [r]$ and $X \in \Delta_r$. Since $\phi^{(r)}$ is an optimal solution for $MR(\Delta_r)$, $X \in \Delta_r$ and $R(X) = 0$ we have that:

$$0 \leq R(X(\phi^{(r)})) = v(MR(\Delta_r)) \leq R(X) = 0$$

With this new approach $\phi^{(k)}$ may be found once and then when c becomes known we can solve $P(c, X(\phi^{(k)}))$ instead of $P(c, X)$, yielding better performance than the usual absolute robust solution obtained with the min max approach. In practice we expect a good performance of $X(\phi^{(k)})$ with k slightly larger than $\|p_r(x_R)\|_1$.

3.2. (q, p, K) -m problem and the use of the Δ_k -case presented

Let $q \geq 1$, $p \in [q]$ and $K \in [p]$. We use a (q, p, K) -m problem to illustrate the use of the Δ_k -case presented.

Let $P(c, X)$ be a (q, p, K) -m problem with q locations, p medians and coverage K . In this case: $n = q + q^2$, $x \in \{0, 1\}^n$ with $x_i = 1$ if and only if a median is located at location i with cost $c_i = 0$ for all $i \in [q]$ and $x_{iq+j} = 1$ if and only if the location j is assigned to a median located at i for all $i, j \in [q]$ with cost c_{iq+j} . Each location must to be assigned to K medians. Let $l, u \in \mathbb{R}^n$ and let us suppose that $0 \leq l \leq c \leq u$ with $l_i = u_i = 0 \ \forall i \in [q]$.

$P(c, X)$ admits a 0-1-MILP formulation as follows:

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

with

$$X = \{x \in \{0, 1\}^n : \sum_{i \in [q]} x_{iq+j} = K \ \forall j \in [q], \sum_{i \in [q]} x_i = p, x_{iq+j} \leq x_i \ \forall i, j \in [q]\}$$

Note that if $K = 1$ then a (q, p, K) -m problem is a classical p -medians problem ([11]).

As an application, imagine an emergency service designed based on the (q, p, K) -m problem. Each time that changes the current situation a new set of p medians could be computed. Even if the computational effort is not large (that is the case for the (q, p, K) -m problems considered in our computational experience) an excessive number of different medians locations over the time may be unacceptable for human users.

Note that $\|p_r(x_R)\|_1 = p$. Let $r = q$ and let $k \in [p, q]$ then:

- $\Phi_k = \{\phi \in \{0, 1\}^q : \|\phi\|_1 = k, X(\phi) \neq \emptyset\} = \{\phi \in \{0, 1\}^q : \|\phi\|_1 = k\}$
- If $\phi \in \Phi_k$ then $P(c, X(\phi))$ is a problem with k (instead of q) potential locations for the medians as follows:

$$P(c, X(\phi)) : \min_{x \in X(\phi)} c^t x$$

with

$$X(\phi) = \{x \in X : x_i \leq \phi_i \forall i \in [q]\}$$

- Each time that c becomes known we solve $P(c, X(\phi^{(k)}))$. The p locations for the medians over the time are taken from a set with k locations (instead of q). In practice we expect a good performance of $X(\phi^{(k)})$ with k slightly larger than p and quite lesser than q .

4. Greedy heuristic to lead with $MR(\Delta_k)$

For the sake of completeness we present in Appendix B an optimal algorithm to solve $MR(\Delta)$ with the framework presented. The algorithm may be seen as a generalization of an algorithm to find an absolute robust solution and iterates between problems to find upper and lower bounds until the gap is closed. Unfortunately the problem to generate the lower bound needs many copies of the 0-1-MILP formulation for $X(\phi)$ (one copie for each solution previously generated) and that may be a huge task for many problems. For that we consider heuristic approaches for $MR(\Delta_k)$.

4.1. Primary algorithm

If $\phi \in \{0, 1\}^r$ let $I(\phi) = \{i : \phi(i) = 1, i \in [r]\}$.

Let $k \in [\|p_r(x_R)\|_1, r]$, let $\phi \in \{0, 1\}^r$ with $\|\phi\|_1 \geq \|p_r(x_R)\|_1$, let $I(\phi)^c = [r] - I(\phi)$. Let $i \in I(\phi)^c$, let $\phi^{+,i} \in \{0, 1\}^r$ such that $I(\phi^{+,i}) = I(\phi) \cup \{i\}$, and let $S(\phi)$ be a problem in (i) defined as follows:

$$S(\phi) : \min_{i \in I(\phi)^c} R(X(\phi^{+,i}))$$

Note that with $S(\phi)$ we are looking for the best $i \in I(\phi)^c$ to be inserted to define the next ϕ and a Greedy algorithm may be designed.

Greedy Algorithm to deal with $MR(\Delta_k)$ (GA):

Let $k \in [\|p_r(x_R)\|_1, r]$, Let $x^1 \in X$ with $\|p_r(x^1)\|_1 = \|p_r(x_R)\|_1$ and let $\phi^1 = p_r(x^1)$. Let $\phi = \phi^1$

1. If $\|\phi\|_1 = k$ stop
2. Solve $S(\phi)$ and let i be an optimal solution.
3. Let $\phi = \phi^{+,i}$ and return to step 1

Let ϕ the output of GA. We have some remarks about ϕ :

- Since $I(\phi^1) \subseteq I(\phi)$ we have that $\{x^1\} \subseteq X(p_r(x^1)) \subseteq X(\phi)$ and then

$$v(S(\phi)) = R(X(\phi)) \leq R(X(p_r(x^1))) \leq R(x^1)$$

$$v(P(c, X(\phi))) \leq v(P(c, X(p_r(x^1)))) \leq c^t x^1 \quad \forall c \in \Omega$$

- If k is *large enough* then $I(p_r(x_R)) \subseteq I(\phi)$ and we have that $\{x_R\} \subseteq X(p_r(x_R)) \subseteq X(\phi)$ and then

$$v(S(\phi)) = R(X(\phi)) \leq R(X(p_r(x_R))) \leq R(x_R)$$

$$v(P(c, X(\phi))) \leq v(P(c, X(p_r(x_R)))) \leq c^t x_R \quad \forall c \in \Omega$$

In practice we expect that $X(\phi)$ has better performance than x_R even if $I(p_r(x_R)) \not\subseteq I(\phi)$

- If $\phi^1 = p_r(x_R)$ then

$$v(S(\phi)) = R(X(\phi)) \leq R(X(p_r(x_R))) \leq R(x_R)$$

$$v(P(c, X(\phi))) \leq v(P(c, X(p_r(x_R)))) \leq c^t x_R \quad \forall c \in \Omega$$

4.2. Secondary algorithm

Now we need an algorithm to solve $S(\phi)$. Note that $S(\phi)$ is a set of $|I(\phi)^c| = r - \|\phi\|_1$ independent $Q1$ -problems: $Q1(X(\phi^{+,i}))$ ($i \in I(\phi)^c$), and then if $k = p_r(x_R) + p^+$ and we use force brute we need to solve $(r - p_r(x_R)) + (r - (p_r(x_R) + 1)) + \dots + (r - (p_r(x_R) + p^+ - 1)) = p^+(r - p_r(x_R)) + \frac{p^+(p^+ - 1)}{2}$ independent $Q1$ -problems to execute GA. However, an algorithm that finds lower and upper bounds until the gap is closed solves $S(\phi)$ solving in general few $Q1$ -problems.

We have an upper bound for $v(S(\phi))$ as follows:

Let $\phi \in \Phi_k$ and let $i \in I(\phi)^c$ then $v(S(\phi)) \leq v(Q1(X(\phi^{+,i})))$

In the next lemma we define a problem to find a lower bound.

Lemma 5. Let $W = \{x^{(1)}, \dots, x^{(s)}\} \subseteq X$. Let $U \subseteq I(\phi)^c$ and let $U^c = I(\phi)^c - U$

Let $w \in [s]$, let $i \in I(\phi)^c$ and let $P(c^+(x^{(w)}), \phi, i)$ be a problem in (x) defined as follows:

$$\begin{aligned} P(c^+(x^{(w)}), \phi, i) : \min & c^+(x^{(w)})^t x \quad s.t \\ & x_j \leq \phi_j^{+,i} \quad \forall j \in [r] \\ & x \in X \end{aligned}$$

Let $S(\phi, U, W)$ a (trivial) problem in (i) defined as follows:

$$\min_{i \in U^c} \left\{ \max_{w \in [s]} \{v(P(c^+(x^{(w)}), \phi, i)) - l^t x^{(w)}\} \right\}$$

then

$$v(S(\phi, U, W)) \leq v(S(\phi))$$

Proof:

$$\begin{aligned} v(S(\phi)) &\stackrel{(1)}{=} \\ \min_{i \in U^c} v(Q1(X(\phi^{+,i}))) &\stackrel{(2)}{=} \min_{i \in U^c} \left\{ \max_{x \in X} \left\{ \min_{y \in X^{(1)}(\phi^{+,i})} c^+(x)^t y - l^t x \right\} \right\} \stackrel{(3)}{\geq} \\ \min_{i \in U^c} \left\{ \max_{x \in W} \left\{ \min_{y \in X^{(1)}(\phi^{+,i})} c^+(x)^t y - l^t x \right\} \right\} &\stackrel{(4)}{=} \\ \min_{i \in U^c} \left\{ \max_{w \in [s]} \left\{ \min_{y \in X^{(1)}(\phi^{+,i})} c^+(x^{(w)})^t y - l^t x^{(w)} \right\} \right\} &\stackrel{(5)}{=} \end{aligned}$$

$$\min_{i \in U^c} \left\{ \max_{w \in [s]} \{v(P(c^+(x^{(w)}), \phi, i)) - l^t x^{(w)}\} \right\} \stackrel{(6)}{=} v(S(\phi, U, W))$$

with:

- (1) From the definition of $S(\phi)$
- (2) From lemma 1
- (3) Since $W \subseteq X$
- (4) From the definition of W
- (5) From the definition of $P(c^+(x^{(w)}), \phi, i)$
- (6) From the definition of $S(\phi, U, W)$ •

Now an algorithm that iterates between $S(\phi, U, W)$ and $Q1(X(\phi^+, i^*))$ until the gap is closed may be defined.

S - $Q1$ -Algorithm (S - $Q1$ -A) to solve $S(\phi)$:

Let $x \in X$ and let $W = \{x\}$, let $U = \emptyset$ and let $UB = \infty$

1. Solve $S(\phi, U, W)$ and let i^* be an optimal solution
2. Solve $Q1(X(\phi^+, i^*))$, let x^* be an optimal solution, let $U = U \cup \{i^*\}$, and let $UB = \min\{UB, R(X(\phi^+, i^*))\}$
3. If $U = I(\phi)^c$ stop
4. If $x^* \in W$ stop
5. Let $W = W \cup \{x^*\}$ and return to step 1

Lemma 6. S - $Q1$ -A stops in a finite number of iterations with $UB = v(S(\phi))$

Proof: If the algorithm stops with $U = I(\phi)^c$ then $UB = \min_{i \in I(\phi)^c} R(X(\phi^+, i)) = v(S(\phi))$.

Let us suppose that i^* is an optimal solution for $S(\phi, U, W)$ (see step 1), let us suppose that x^* is an optimal solution for $Q1(X(\phi^+, i^*))$ (see step 2) and let us suppose that $x^* \in W$ then:

$$\begin{aligned} \min_{i \in U^c} R(X(\phi^+, i)) &\stackrel{(1)}{\geq} v(S(\phi, U, W)) \stackrel{(2)}{=} \max_{w \in [s]} \min_{y \in X(\phi^+, i^*)} c^+(x^{(w)})^t y - l^t x^{(w)} \stackrel{(3)}{\geq} \\ &\min_{y \in X(\phi^+, i^*)} c^+(x^*)^t y - l^t x^* \stackrel{(4)}{=} v(Q1(X(\phi^+, i^*))) \stackrel{(5)}{\geq} UB \stackrel{(6)}{=} \min_{i \in U} R(X(\phi^+, i)) \end{aligned}$$

with:

- (1) Since $v(S(\phi, U, W))$ is a lower bound

- (2) Since i^* is an optimal solution for $S(\phi, U, W)$
- (3) Since $x^* \in W$
- (4) Since x^* is an optimal solution for $Q1(X(\phi^+, i^*))$
- (5) and (6) From step 2 of the algorithm

and then $UB = \min_{i \in I(\phi)^c} R(X(\phi^+, i))$ •

5. Computational experience

Now we present our computational experience with GA using S - $Q1$ -A to deal with $MR(\Delta_k)$ for (q, p, K) -m problems.

5.1. Computational setup

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

- Intel(R)Core(TM) i7-9750H CPU, @ 2.60 GHz Lenovo ThinkPad X1 Extreme Gen 2, 32.00 GB Ram and Windows 10 Pro Operating System
- All the instances have been processed through ILOG-Cplex 12.10 (<http://ibm-ilog-cplex-optimization-studio-acade.software.informer.com/12.10/>) from a MATLAB code
- All the parameters of Cplex are in their default values with the exemption of: the relative tolerance parameter *tolerance.mipgap* that was set to 0 and the parameter to declare that a variable took the value 0 or 1 (*tolerance.integrality*) that was set to 0

5.2. Data for the (q, p, K) -m problems

The data were generated at random using a standard procedure ([6]) as follows:

- let (a_j, b_j) be the location j taken from $U((0, 100) \times (0, 100)) \forall j \in [q]$
- let D_j be the demand of location j taken from $U(0, 100) \forall j \in [q]$
- let d_{ij} be the distance from location i to location j computed as $d_{ij} = |a_i - a_j| + |b_i - b_j| \forall i, j \in [q]$
- let $L_{ij} = d_{ij}D_j \forall i, j \in [q]$
- let $\alpha, \beta \in (0, 1]$, let s_{ij} taken from $U(0, 1)$, if $s_{ij} < \beta$ then let $U_{ij} = (1 + r_{ij}\alpha)L_{ij}$ where r_{ij} is taken from $U(0, 1)$ and if $s_{ij} \geq \beta$ let $U_{ij} = L_{ij} \forall i, j \in [q]$
- let $l_i = u_i = 0 \forall i \in [q]$
- let $l_{iq+j} = L_{ij}$ and let $u_{iq+j} = U_{ij} \forall i, j \in [q]$

5.3. Parameters reported in tables

Let p, p^+, K, q, α and β the parameters to define the (q, p, K) -m problems and the $MR(\Delta_k)$ problems with $k = p + p^+$ as indicated.

Data indexed with $set \in \{1, \dots, 8\}$ were generated with (q, p, K) as indicated. We use $\alpha, \beta \in \{0.50, 0.75, 1.00\}$. For each set and for each pair (α, β) we generate 10 (l, u) -cases, therefore we have 90 (l, u) -cases for each set . For each (l, u) we generate 100 c -cases with $c \in \Omega = \{c \in \mathbb{R}^n : l \leq c \leq u\}$, therefore we have 9000 $P(c, X)$ problems for each set . We use $q \in \{100, 150\}$, $p \in \{5, 10\}$ and $K \in \{1, 2\}$.

Let $LU(set)$ the set of 90 (l, u) -cases generated for set . Let $C(l, u)$ the set of 100 c -cases generated for (l, u) .

Let $x(l)$ be an optimal solution for $P(l, X)$. Let $x_R^{l,u}$ be an absolute robust solution for the (l, u) -case. Let $\phi^1 \in \{p_r(x_R^{l,u}), p_r(x(l))\}$ and let $x^1 \in \{x_R^{l,u}, x(l)\}$. For each (l, u) generated let $\phi_{p^+}^{l,u}(\phi^1)$ be the solution generated with GA with $k = p + p^+$ starting with $\phi = \phi^1$.

5.4. Values reported in tables

Values reported in tables are either averages or maximal values of some statistics computed for the 90 (l, u) -cases generated for each set and the 100 c -cases generated for each (l, u) as follows:

5.4.1. Times and iterations

- Let $t_R^{l,u}$ be the time to find $x_R^{l,u}$ using a standard algorithm ([12])

For each set as indicated we report

$$\overline{t_R}(set) = \frac{1}{90} \sum_{(l,u) \in LU(set)} t_R^{l,u}$$

and

$$\widehat{t_R}(set) = \max\{t_R^{l,u} : (l, u) \in LU(set)\}$$

- Let $t_{p^+}^{l,u}(\phi^1)$ be the time to find $\phi_{p^+}^{l,u}(\phi^1)$. If $\phi^1 = p_r(x_R^{l,u})$ then $t_{p^+}^{l,u}(\phi^1)$ includes the time to find $x_R^{l,u}$ and if $\phi^1 = p_r(x(l))$ then $t_{p^+}^{l,u}(\phi^1)$ includes the time to find $x(l)$.

For each set, p^+ and ϕ^1 as indicated we report:

$$\bar{t}(set, p^+, \phi^1) = \frac{1}{90} \sum_{(l,u) \in LU(set)} t_{p^+}^{l,u}(\phi^1)$$

and

$$\widehat{t}(set, p^+, \phi^1) = \max \{t_{p^+}^{l,u}(\phi^1) : (l, u) \in LU(set)\}$$

- Let $t_*(c)$ be the time to solve $P(c, X)$ to optimality and let $to(\phi_{p^+}^{l,u}(\phi^1))$ the time to solve $P(c, X(\phi_{p^+}^{l,u}(\phi^1)))$ to optimality.

For each set, p^+ and ϕ^1 as indicated we report the maximal percentage of the total time to solve the 100 problems of each (l, u) -case using $X(\phi_{p^+}^{l,u}(p_r x_R))$ relative to the time using X to solve the same problems as follows:

$$\widehat{tr}(set, p^+, \phi^1) = 100 \times \max \left\{ \frac{\sum_{c \in C(l,u)} to(\phi_{p^+}^{l,u}(\phi^1))}{\sum_{c \in C(l,u)} t_*(c)} : (l, u) \in set \right\}$$

- Let $i_{p^+}^{l,u}(\phi^1)$ be the number of iterations to solve the sequence of $S(\phi)$ -problems in order to find $\phi_{p^+}^{l,u}(\phi^1)$.

For each set, p^+ and ϕ^1 as indicated we report the average and the maximal percentage of iterations used relative to the number of iterations using brute force to solve the sequence of $S(\phi)$ -problems, as follows:

$$\bar{i}(set, p^+, \phi^1) = 100 \times \frac{\frac{1}{90} \sum_{(l,u) \in LU(set)} i_{p^+}^{l,u}(\phi^1)}{p^+(q-p) + \frac{p^+(p^+-1)}{2}}$$

and

$$\widehat{i}(set, p^+, \phi^1) = 100 \times \frac{\max \{i_{p^+}^{l,u}(\phi^1) : (l, u) \in LU(set)\}}{p^+(q-p) + \frac{p^+(p^+-1)}{2}}$$

- Let $iQ_{p^+}^{l,u}(\phi^1)$ be the number of $Q2$ -problems solved using $Q2$ - P -A to solve the sequence of $Q1$ - problems (see step 2 of S - $Q1$ -A) in order to find $\phi_{p^+}^{l,u}(\phi^1)$

For each set, p^+ and ϕ^1 as indicated we report

$$\bar{iQ}(set, p^+, \phi^1) = \frac{1}{90} \sum_{(l,u) \in LU(set)} iQ_{p^+}^{l,u}(\phi^1)$$

and

$$\widehat{iQ}(set, p^+, \phi^1) = \max \{iQ_{p^+}^{l,u}(\phi^1) : (l, u) \in LU(set)\}$$

5.4.2. *Regret reduction and relative errors*

- Let $\delta_{p^+}^{l,u}(\phi^1, x^1)$ be the regret reduction percentage of $X(\phi_{p^+}^{l,u}(\phi^1))$ with respect to $R(x^1)$, computed as follows:

$$\delta_{p^+}^{l,u}(\phi^1, x^1) = 100 \times \frac{R(x^1) - R(X(\phi_{p^+}^{l,u}(\phi^1)))}{R(x^1)}$$

with $\phi_0^{l,u}(\phi^1) = \phi^1$.

For each set, p^+, ϕ^1 and x^1 as indicated we report

$$\bar{\delta}_{p^+}(set, \phi^1, x^1) = \frac{1}{90} \sum_{(l,u) \in LU(set)} \delta_{p^+}^{l,u}(\phi^1, x^1)$$

- Let $\mu_{p^+}^{l,u}(\phi^1)$ be the average of relative errors percentages for $X(\phi_{p^+}^{l,u}(\phi^1))$ over $C(l, u)$, computed as follows:

$$\mu_{p^+}^{l,u}(\phi^1) = \frac{1}{|C(l, u)|} \sum_{c \in C(l, u)} 100 \times \frac{v(P(c, X(\phi_{p^+}^{l,u}(\phi^1)))) - v(P(c, X))}{v(P(c, X))}$$

Since $|C(l, u)| = 100$ then $\mu_{p^+}^{l,u}(\phi^1)$ may be interpreted as the relative errors percentages sum over $C(l, u)$

For each set, p^+ and ϕ^1 we report:

$$\bar{\mu}_{p^+}(set, \phi^1) = \frac{1}{90} \sum_{(l,u) \in LU(set)} \mu_{p^+}^{l,u}(\phi^1)$$

- Let $\mu^{l,u}(x^1)$ be the average of relative errors percentages for x^1 over $C(l, u)$, computed as follows:

$$\mu^{l,u}(x^1) = \frac{1}{|C(l, u)|} \sum_{c \in C(l, u)} 100 \times \frac{c^t x^1 - v(P(c, X))}{v(P(c, X))}$$

Since $|C(l, u)| = 100$ then $\mu^{l,u}(x^1)$ may be interpreted as the relative errors percentages sum for x^1 over $C(l, u)$

Let $\gamma_{p^+}^{l,u}(\phi^1, x^1)$ be the average of relative errors reduction percentages of $X(\phi_{p^+}^{l,u}(\phi^1))$ with respect to $\mu^{l,u}(x^1)$ computed as follows:

$$\gamma_{p^+}^{l,u}(\phi^1, x^1) = 100 \times \frac{\mu^{l,u}(x^1) - \mu_{p^+}^{l,u}(\phi^1)}{\mu^{l,u}(x^1)}$$

For each set, p^+ and ϕ^1 we report:

$$\bar{\gamma}_{p^+}(set, \phi^1) = \frac{1}{90} \sum_{(l,u) \in C(l,u)} \gamma_{p^+}^{l,u}(\phi^1, x^1)$$

5.4.3. Ratio of medians used with optimal solutions

- Let $mo(c)$ be the set of medians to solve $P(c, X)$ to optimality. For each (l, u) let:

$$Mo(l, u) = \bigcup_{c \in C(l, u)} mo(c)$$

$Mo(l, u)$ is the set of all medians locations to solve $P(c, X)$ to optimality $\forall c \in C(l, u)$. Note that we use $p + p^+$ different location medians to solve $P(c, X(\phi^{l,u}(\phi^1)))$ for all $c \in C(l, u)$.

For each set and p^+ as indicated we report the average and the maximal value of the ratio of medians used with optimal solutions as follows:

$$\overline{mo}(set, p^+) = \frac{\frac{1}{90} \sum_{(l,u) \in set} |Mo(l, u)|}{p + p^+}$$

and

$$\widehat{mo}(set, p^+) = \frac{\max\{|Mo(l, u)| : (l, u) \in set\}}{p + p^+}$$

5.5. Performance of GA to deal with $MR(\Delta_k)$ and remarks about the generated solutions

In table 1 we report times and iterations for 8 *sets* with (q, p, K) as indicated and starting with $\phi^1 = p_r(x_R^{l,u})$. The experimentation for *sets* 7 and 8 is not reported since $t_R^{l,u}$ exceeded the predefined time limit (24000 seconds) several times using a standard algorithm and the experimentation was stopped. We will consider *sets* 7 and 8 later in this section.

It can be seen that \bar{i} and \hat{i} were less than 6.3% and 7.9% for the problems in *sets* 1 to 4 respectively and less than 7.1% and 11.7% for the problems in *sets* 5 and 6 respectively. \bar{iQ} and \hat{iQ} also kept at very low values with an atypical case in *set* 4 that required more than a thousand iterations.

\bar{t} and \hat{t} were also kept at tolerable values, below 1215 and 3794 seconds (s.) for the problems of *sets* 1 to 5 respectively (note that *GA* it runs offline and then we use the solutions in *real time*). The same goes for *set* 6 except for the

presence of an atypical case that required more than 16000 s. to find $x_R^{l,u}$.

By completeness $\widehat{t_r}$ is presented even though in these experiments that statistic is not so important because the problems in X are solved in seconds for the dimensions considered.

It is important to note in table 1 that the time to find $x_R^{l,u}$ for *sets* 1 to 6 is, on average, about the 56% of the time required by the algorithm, with some cases over 90%. That justifies the idea of evaluating what happens if instead of starting from $p_r(x_R^{l,u})$ is done from $p_r(x(l))$ (any x in X may be used) slightly increasing p^+ . The quality of the solutions would be offset by the increase in p^+ .

In table 2 we report regret reduction, relative errors and ratios of medians used starting with $p_r(x_R^{l,u})$ and with $x^1 = x_R^{l,u}$. We can see δ_0 and δ_{p^+} with values ranging from 0.0% to 0.4% and from 3.1% to 11.8% respectively (the maximum values not reported in the table reached up to 75%). Also, we present $\overline{\gamma}_0$ and $\overline{\gamma}_{p^+}$.

The most important highlight in table 2 is the relative error average that was achieved using the solutions in $X(\phi_{p^+}^{l,u}(p_r x_R))$ instead of the optimal solutions, which was below 1% for *sets* 1 through 6. That level of error does not seem extraordinary (heuristics with similar relative errors abound for several combinatorial problems) except for the significant fact that optimal solutions use on average from 3 to 4 times the number of locations used with the solutions in $X(\phi_{p^+}^{l,u}(p_r x_R))$, and up to 7 times in the worst case.

In tables 3, 4 and 5 we compare the behavior of the algorithm and the solutions obtained starting with $p_r(x_R^{l,u})$ and $p_r(x(l))$.

In tables 3 and 4 we report regret reduction and times as p^+ grows when starting from $p_r(x_R^{l,u})$ and $p_r(x(l))$ (any other x could be used) and with $x^1 = x_R^{l,u}$.

As expected, regret reduction starting from $p_r(x(l))$ starts at negative values and increases as p^+ grows.

For *sets* 1,2, and 3 the regret reduction obtained starting with $p_r(x(l))$ and using $p^+ = 4$ improved the one obtained by starting with $p_r(x_R^{l,u})$ using $p^+ = 2$. For *set* 6 the regret reduction obtained by starting with $p_r(x(l))$ and using $p^+ = 6$ improve the one obtained starting with $p_r(x_R^{l,u})$ using $p^+ = 4$. For *sets* 4 and 5 the regret reduction starting with $p_r(x(l))$ reached levels comparable to those obtained starting from $p_r(x_R^{l,u})$.

In the same tables you can see the evolution of the execution time as p^+ grows for both strategies. Unsurprisingly the total time starting from $p_r(x(l))$ was significantly less than the time used starting from $p_r(x_R^{l,u})$ for the same p^+ .

Table 1: Times and iterations with $\phi^1 = p_r(x_R^{l,u})$

<i>set</i>	<i>p</i>	<i>K</i>	<i>q</i>	p^+	$\overline{t_R}$	$\widehat{t_R}$	\bar{t}	\hat{t}	\bar{i}	\hat{i}	\overline{iQ}	\widehat{iQ}	\widehat{tr}
1	5	1	100	2	43	144	120	238	5.9	7.9	26.6	52	14.6
2			150		195	589	413	841	3.8	4.8	25.5	43	7.2
3		2	100		205	928	298	1031	6.3	7.3	27.9	47	11.8
4			150		950	3490	1215	3794	4.2	4.8	27.1	51	5.1
5	10	1	100	4	226	1324	549	1667	7.1	11.7	161.7	1084	24.4
6			150		1691	16582	2625	17821	5.8	8.0	142.4	382	14.5
7		2	100		∞								
8			150		∞								

Table 2: Regret reduction, relative errors and ratios of medians used with $(\phi^1, x^1) = (p_r(x_R^{l,u}), x_R^{l,u})$

<i>set</i>	p^+	$\bar{\delta}_0$	$\bar{\delta}_{p^+}$	$\bar{\gamma}_0$	$\bar{\gamma}_{p^+}$	$\bar{\mu}_{p^+}$	\bar{m}_0	\bar{m}_0
1	2	0.1	8.7	36.9	50.2	0.8	3.2	6.0
2		0.0	5.3	39.8	50.7	0.7	3.5	6.4
3		0.0	5.0	51.8	63.3	0.6	3.4	6.3
4		0.0	3.1	55.1	61.9	0.6	4.0	7.0
5	4	0.4	7.1	32.7	52.0	0.7	2.4	3.8
6		0.1	11.8	33.1	47.8	0.9	3.0	4.9

In table 5 we compare relative errors, ratios of median used and times with $x^1 = x_R^{l,u}$ for the two strategies using the corresponding maximum p^+ . Note that the averages of the relative errors are similar with slight advantage for the $p_r(x(l))$ strategy, this at the cost of slightly decreasing the ratio of medians used and increasing times to obtain optimal solutions. The conclusion that can be inferred from tables 3,4 and 5 is that if time to obtain $\phi_{p^+}^{l,u}(\phi^1)$ becomes a problem using $p_r(x_R^{l,u})$ it can be started from $p_r(x(l))$ by slightly increasing the potential number of medians to be used, p^+ , without that causing a deterioration of the quality of the solutions and the time.

Finally, tables 6,7 and 8 refers to *sets* 7 and 8 starting from $p_r(x(l))$. Now GA was capable to generate solutions in tolerable times. Again, the most important highlight may be seen in in Table 7: the relative errors average that was achieved using the solutions in $X(\phi_{p^+}^{l,u}(p_r x(l)))$ instead of the optimal solutions, $\bar{\mu}_{p^+}$, are below 1% for *sets* 7 and 8 with the ratio of medians used with optimal solutions over 3.4

Table 3: Regret reduction and times as p^+ grows with $x^1 = x_R^{l,u}$, $\phi^{1,R} = p_r(x_R^{l,u})$ and $\phi^{1,l} = p_r(x(l))$

set	ϕ^1	$\phi^{1,R}$			$\phi^{1,l}$				
	p^+	0	1	2	0	1	2	3	4
1	$\bar{\delta}$	0.1	5.3	8.7	-21.7	-6.8	1.2	7.0	10.5
	\bar{t}	44	80	120	1	36	77	124	180
2	$\bar{\delta}$	0.0	3.1	5.3	-19.3	-7.7	-0.7	3.3	6.1
	\bar{t}	198	298	413	4	108	230	382	559
3	$\bar{\delta}$	0.0	2.8	5.0	-17.7	-6.3	0.1	3.8	6.5
	\bar{t}	206	249	298	1	66	140	225	329
4	$\bar{\delta}$	0.0	1.9	3.1	-16.5	-8.0	-2.9	0.8	3.0
	\bar{t}	953	1073	1215	4	122	264	427	628

Table 4: Regret reduction and times as p^+ grows with $x^1 = x_R^{l,u}$ and $\phi \in \{\phi^{1,R}, \phi^{1,l}\}$ with $\phi^{1,R} = p_r(x_R^{l,u})$, and $\phi^{1,l} = p_r(x(l))$

set	ϕ^1	$\phi^{1,R}$						$\phi^{1,l}$						
	p^+	0	1	2	3	4	0	1	2	3	4	5	6	
5	$\bar{\delta}$	0.4	6.9	11.3	14.4	14.4	-25.2	-11.8	-2.5	4.2	9.6	13.7	17.3	
	\bar{t}	227	285	353	437	549	1	62	134	221	334	489	696	
6	$\bar{\delta}$	0.1	4.6	7.6	9.8	9.8	-24.7	-13.2	-5.9	-0.4	3.7	6.7	9.2	
	\bar{t}	1693	1866	2062	2303	2625	3	152	332	547	821	1150	1597	

Table 5: Relative errors, ratios of medians used and times with $x^1 = x_R^{l,u}$

set	ϕ^1	$\bar{\mu}$		\overline{mo}		\bar{tr}	
		$p_r(x_R^{l,u})$	$p_r(x(l))$	$p_r(x_R^{l,u})$	$p_r(x(l))$	$p_r(x_R^{l,u})$	$p_r(x(l))$
	p^+	2	4	2	4	2	4
1		0.8	0.6	3.2	2.5	11.0	14.1
2		0.7	0.5	3.5	2.7	4.1	4.9
3		0.6	0.4	3.4	2.7	8.5	15.5
4		0.6	0.4	4.0	3.1	3.4	4.3
	p^+	4	6	4	6	4	6
5		0.7	0.7	2.4	2.0	19.2	22.5
6		0.9	0.8	3.0	2.6	9.2	10.5

Table 6: Times and iterations with $\phi^1 = (p_r(x(l)))$

<i>set</i>	<i>p</i>	<i>K</i>	<i>q</i>	p^+	\bar{t}	\hat{t}	\bar{i}	\hat{i}	\bar{iQ}	\widehat{iQ}	\hat{tr}
7	10	2	100	6	1739	9062	13.7	17.8	493.1	1631	33.0
8			150		2732	8463	8.9	12.0	327.0	869	9.8

Table 7: Regret reduction, relative errors and ratios of medians used with $(\phi^1, x^1) = (p_r(x(l)), x(l))$

<i>set</i>	p^+	$\bar{\delta}_0$	$\bar{\delta}_{p^+}$	$\bar{\gamma}_0$	$\bar{\gamma}_{p^+}$	$\bar{\mu}_{p^+}$	\bar{m}_0	\widehat{m}_0
7	6	11.9	35.0	41.5	84.0	0.4	2.0	3.4
8		11.3	29.9	39.4	81.8	0.5	2.6	4.6

6. Conclusions and further extensions

6.1. Conclusions

- The traditional absolute robustness approach is to find a single absolute robust solution by solving a min max problem. In the search for better quality solutions to face uncertainty, the traditional approach was extended in a previous work finding a set of k solutions solving a min max min absolute regret problem.

In this paper we have proposed one more level when considering the absolute regret of a set implicitly defined based on some practical interest property and the corresponding *new* min max min absolute regret problem was considered.

- An optimal algorithm to compute the absolute regret was presented and a greedy heuristic to deal with the min max min absolute regret problem for a practical interest case based on bounded projections was presented.
- Experiments for (q, p, K) -m problems were presented. The computational effort to find the final solution set may be considerable however is important to note that the greedy algorithm is executed offline. The idea is that once the final solution set is found, simpler problems than the original

Table 8: Regret reduction and times as p^+ grows with $(\phi^1, x^1) = (p_r(x(l)), x(l))$

<i>set</i>	p^+	0	1	2	3	4	5	6
7	$\bar{\delta}$	11.9	19.1	24.0	27.8	30.7	33.0	35.0
	\bar{t}	2	97	212	367	606	999	1740
8	$\bar{\delta}$	11.3	16.9	20.8	23.8	26.3	28.1	29.9
	\bar{t}	7	232	501	842	1316	1893	2733

may be solved in tolerable times when c becomes known obtaining better quality solutions than with the previous approaches. Note that the relative error average for the solutions found was below 1% for the problems considered.

- The most significant fact is that the solutions found have the practical interest property considered. In the (q, p, K) -m problem the final set based on bounded projections restricts the search of the p medians by considering k locations with k significantly less than q and this may be important for the human users.

6.2. Further extensions

Our next steps are: (i) design an heuristic algorithm to deal with the case based on neighborhoods (see Appendix A) to be applied to (q, p, K) -m problems and shortest path problems on graphs and (ii) extend the theory and the algorithms to the relative regret robustness case.

Declaration of Competing Interest

None

References

- [1] A. Chassein, M. Goerigk, Performance analysis in robust optimization, Doumpos M., Zopounidis C., Grigoroudis E. (eds) Robustness Analysis in Decision Aiding, Optimization, and Analytics. International Series in Operations Research and Management Science, Springer 241 (2016).
- [2] C. Buchheim, J. Kurtz, Robust combinatorial optimization under convex and discrete cost uncertainty, EURO Journal on Computational Optimization 6 (3) (2018) 211–238.
- [3] J. Li, Y. Liu, Approximation algorithms for stochastic combinatorial optimization problems, J. Oper. Res. Soc. China 4 (2016) 1–47.
- [4] R. Oberdieck, N. Diangelakis, I. Nascu, M. Papathanasiou, S. M., S. Avraamidou, P. E.N., On multi-parametric programming and its applications in process systems engineering, Chem. Eng. Res. Des. 116 (2016) 61–82.
- [5] C. Carlsson, R. Fuller, Fuzzy reasoning in decision making and optimization, Studies in Fuzziness and Soft Computing (2002). doi:<https://doi.org/10.1007/978-3-7908-1805-5>.
- [6] A. Crema, Min max min robust (relative) regret combinatorial optimization, Math Meth Oper Res 92 (2020) 249–283. doi:<https://doi.org/10.1007/s00186-020-00712-y>.

- [7] C. Buchheim, J. Kurtz, Min-max-min robustness: a new approach to combinatorial optimization under uncertainty based on multiple solutions, *Electronic Notes in Discrete Mathematics* 52 (2016) 45–52.
- [8] C. Buchheim, J. Kurtz, Min–max–min robust combinatorial optimization, *Mathematical Programming* 163 (1-2) (2017) 1–23.
- [9] J. Kurtz, New complexity results and algorithms for min-max-min robust combinatorial optimization, *arXiv:2106.03107v2 [math.OC]* (2021).
- [10] M. Karatasa, N. Razib, H. Tozanc, A comparison of p-median and maximal coverage location models with q coverage requirement, *Procedia Engineering* 149 (2016) 169–176.
- [11] T. B. Boffey, J. Karkazis, p-medians and multi-medians, *Journal of the Operational Research Society* 35 (1) (1984) 57–64. doi:10.1057/jors.1984.6.
- [12] H. Aissic, C. Bazgan, D. Vanderpooten, Min–max and min–max regret versions of combinatorial optimization problems: A survey, *European Journal of Operational Research* 197 (2) (2009) 427–438. doi:https://doi.org/10.1016/j.ejor.2008.09.012.

Appendix A: Δ^k -case based on neighborhoods

Let $r \in [n]$ and let $k \in [r]$, let $x \in X$, let $N_r^k(p_r(x))$ the k -neighborhood of $p_r(x)$ defined as follows:

$$N_r^k(x) = \{y \in X : \|p_r(y) - p_r(x)\|_1 \leq k\}$$

let $\Phi^k = \{\phi \in \{0,1\}^r : \phi = p_r(x) : x \in X\}$ and let $\Delta^k = \{Y : Y = N_r^k(p_r(x)) : x \in X\}$

Let $P(c, X)$ be a (q, p, K) -m problem, let $r = q$ and let $\phi^* = p_r(x^*)$ be an optimal solution for $MR(\Delta^k)$, then each time that c becomes known we solve $P(c, N_r^k(p_r(x^*)))$ instead of $P(c, X)$. The p locations medians over the time are taken in such a manner that we have at most k changes with respect to the p location medians of the reference solution x^* . Note that $R(N_r^k(p_r(x^*))) \leq R(N_r^k(p_r(x_R))) \leq R(x_R)$.

Appendix B: An algorithm to solve the min max min absolute regret problem

The framework considered for $MR(\Delta)$ is defined as follows: let $r \in [n]$, let $\Phi \subseteq \{0,1\}^r$ and if $\phi \in \Phi$ let $X(\phi) \subseteq X$ with $X(\phi) \neq \emptyset$. The sets $X(\phi) \forall \phi \in \Phi$ and the set Φ admit a 0-1-MILP formulation.

The problem is:

$$MR(\Delta) : \min_{Y \in \Delta} R(Y) = \min_{\phi \in \Phi} R(X(\phi))$$

Next we present an optimal algorithm to solve $MR(\Delta)$ that works for all $MR(\Delta)$ with the general framework presented.

The algorithm iterates between problems to compute upper and lower bounds for $\min_{\phi \in \Phi} R(X(\phi))$ until the gap is closed.

Next we present a lemma that includes a problem to compute the lower bound. The lower bound may be computed solving directly a 0-1-MILP problem, however the problem includes many copies of the 0-1-MILP formulation for $X(\phi)$ and that is a huge computational task for many problems.

Lemma 7. *Let $\Psi \subseteq \Omega$ with $\Psi \neq \emptyset$. Let $LB(\Psi)$ be a problem defined in (ϕ) as follows:*

$$LB(\Psi) : \min_{\phi \in \Phi} \max_{c \in \Psi} \left\{ \min_{x \in X(\phi)} c^t x - v(P(c, X)) \right\}$$

then:

$$(i) F(LB(\Psi)) \neq \emptyset$$

$$(ii) v(LB(\Psi)) \leq \min_{\phi \in \Phi} R(X(\phi))$$

$$(iii) \text{ Let } r(\Psi) \in \mathbb{Z} \text{ with } r(\Psi) \geq 1 \text{ and let us suppose that } \Psi = \{c^{(1)}, \dots, c^{(r(\Psi))}\}.$$

$LB(\Psi)$ may be rewritten as a 0-1-MILP problem as follows:

$$\begin{aligned} & \min \sigma \text{ s.t.} \\ & \sigma \geq c^{(i)t} x^{(i)} - v(P(c^{(i)}, X)) \quad \forall i \in [r(\Psi)] \\ & x^{(i)} \in X(\phi) \quad \forall i \in [r(\Psi)], \phi \in \Phi, \sigma \in \mathbb{R} \end{aligned}$$

Proof:

(i) Since $X(\phi) \neq \emptyset \quad \forall \phi \in \Phi$ we have that $F(LB(\Psi)) \neq \emptyset$

(ii) Since $\Psi \subseteq \Omega$ we have that:

$$\begin{aligned} \min_{\phi \in \Phi} R(X(\phi)) &= \min_{\phi \in \Phi} \max_{c \in \Omega} \left\{ \min_{x \in X(\phi)} c^t x - v(P(c, X)) \right\} \geq \\ & \min_{\phi \in \Phi} \max_{c \in \Psi} \left\{ \min_{x \in X(\phi)} c^t x - v(P(c, X)) \right\} = v(LB(\Psi)) \end{aligned}$$

$$(iii) \ v(LB(\Psi)) = \min_{\phi \in \Phi} \max_{c \in \Psi} \left\{ \min_{x \in X(\phi)} c^t x - v(P(c, X)) \right\} =$$

$$\min \sigma \text{ s.t. :}$$

$$\sigma \geq c^{(i)t} x^{(i)} - v(P(c^{(i)}, X)) \quad \forall i \in [r(\Psi)]$$

$$x^{(i)} \in X(\phi) \quad \forall i \in [r(\Psi)], \phi \in \Phi, \sigma \in \mathbb{R}$$

Let $\Psi^{(1)} \neq \emptyset$. Let $\Psi^{(1)} \subseteq \Psi^{(2)} \subseteq \Omega$ then:

$$v(LB(\Psi^{(1)})) \leq v(LB(\Psi^{(2)}))$$

Note that the formulation of $LB(\Psi)$ has $r(\Psi)$ copies of the 0-1-MILP formulation of $X(\phi)$.

$LB(\Psi)$ -Q1-Algorithm ($LB(\Psi)$ -Q1-A) to solve $MR(\Delta)$:

Let $\Psi \subseteq \Omega$ with $\Psi \neq \emptyset$, let $\phi \in \Phi$, let $LB = 0$ and let $UB = R(X(\phi))$.

1. Solve $LB(\Psi)$, let ϕ^* be an optimal solution
2. Solve $Q1(X(\phi^*))$, let $UB = \min\{UB, R(X(\phi^*))\}$, and let x^* be an optimal solution for $Q1(X(\phi^*))$
3. If $c^+(x^*) \in \Psi$ Stop
4. Let $\Psi = \Psi \cup \{c^+(x^*)\}$ and return to step 1

In step 1 we generate a lower bound for $\min_{\phi \in \Phi} R(X(\phi))$ and a new solution $\phi^* \in \Phi$. In step 2 the upper bound is updated. If $c^+(x^*)$ we have an optimal solution. In step 4 Ψ is updated before return to step 1.

Note that solving $LB(\Psi)$ we have that the new solution ϕ has the best absolute regret *with the regret constrained to Ψ* . Also, note that the sequence of lower bound values is non-decreasing.

Next we present a lemma about the optimality of the algorithm. .

Lemma 8. $LB(\Psi)$ -Q1-A stops in a finite number of iterations with $UB = \min_{\phi \in \Phi} R(X(\phi))$

Proof:

Let us suppose that $c^+(x^*) \in \Psi$ then:

$$\begin{aligned} \min_{\phi \in \Phi} R(X(\phi)) &\geq v(LB(\Psi)) = \max_{c \in \Psi} \left\{ \min_{x \in X(\phi^*)} c^t x - v(P(c, X)) \right\} \geq \\ &\min_{x \in X(\phi^*)} c^+(x^*)^t x - v(P(c^+(x^*), X)) = v(Q1(X(\phi^*))) = R(X(\phi^*)) \\ &\geq UB \geq \min_{\phi \in \Phi} R(X(\phi)) \end{aligned}$$

Since X is a finite set then $c^+(x^*) \in \Psi$ in a finite number of iterations.