Regret Robustness with Discrete Uncertainty

Robust Optimization

Stefan Lendl

Institute of Discrete Mathematics
Graz University of Technology

Regret

Strict robust optimization (MIN-MAX): minimize the worst-case cost

"Human evaluation" of solutions often:

Regret

Strict robust optimization (MIN-MAX): minimize the worst-case cost

"Human evaluation" of solutions often:

- Happens after the scenario is revealed.
- Does not care about the uncertainty before making the decision, or what else could have happend.
- Compares to the best possible decision for the true revealed scenario - **the regret**.

Regret

Strict robust optimization (MIN-MAX): minimize the worst-case cost

"Human evaluation" of solutions often:

- Happens after the scenario is revealed.
- Does not care about the uncertainty before making the decision, or what else could have happend.
- Compares to the best possible decision for the true revealed scenario - **the regret**.

Problem: MIN-MAX solution might have terrible regret.

Regret

Strict robust optimization (MIN-MAX): minimize the worst-case cost

"Human evaluation" of solutions often:

- Happens after the scenario is revealed.
- Does not care about the uncertainty before making the decision, or what else could have happend.
- Compares to the best possible decision for the true revealed scenario - **the regret**.

Problem: MIN-MAX solution might have terrible regret.

Solution: Minimize the worst-case regret.

Regret Robustness: Formal Definition

MIN-MAX REGRET

$$\min_{x \in \mathcal{X}} \max_{c \in \mathcal{U}} c(x) - \mathrm{opt}(c)$$

where

$$\mathrm{opt}(c) = \min_{x \in \mathcal{X}} c(x)$$

Regret Robustness: Formal Definition

Min-Max Regret

$$\min_{x\in\mathcal{X}} \max_{c\in\mathcal{U}} c(x) - \text{opt}(c)$$

where

$$\text{opt}(c) = \min_{x\in\mathcal{X}} c(x)$$

We also write

$$r(x,c) = c(x) - \text{opt}(c)$$

and

$$r(x) = \max_{c\in\mathcal{U}} r(x,c)$$

for the regret of x under scenario c and for the worst-case regret of x.

Example: Knapsack

Cash outflows and profits of the investments

i	w_i	p_i^1	p_i^2	p_i^3
1	3	4	3	3
2	5	8	4	6
3	2	5	3	3
4	4	3	2	4
5	5	2	8	2
6	3	4	6	2

1	2	3	Optimal solution
17	10	12	Scenario 1: (1,1,1,0,0,0)
11	17	7	Scenario 2: (0,0,1,0,1,1)
16	9	13	Scenario 3: (0,1,1,1,0,0)
15	12	12	Max–min: (0,1,0,1,0,1)
15	15	11	Min–max regret: (0,1,1,0,1,0)

Computational Complexity: Discrete Uncertainty

number of scenarios K constant

Problems	Constant	
	Min–max	Min–max regret
SHORTEST PATH	*NP*-hard, pseudo-poly [65]	*NP*-hard, pseudo-poly [65]
SPANNING TREE	*NP*-hard [43], pseudo-poly [3]	*NP*-hard [43], pseudo-poly [3]
ASSIGNMENT	*NP*-hard [43]	*NP*-hard [43]
KNAPSACK	*NP*-hard, pseudo-poly [43]	*NP*-hard, pseudo-poly [43]
MIN CUT	Polynomial [7]	Polynomial [2]
MIN s—t CUT	strongly *NP*-hard [2]	strongly *NP*-hard [2]

Computational Complexity: Discrete Uncertainty

number of scenarios K non-constant

Problems	Non-constant	
	Min–max	Min–max regret
SHORTEST PATH	Strongly *NP*-hard [43]	Strongly *NP*-hard [43]
SPANNING TREE	Strongly *NP*-hard [43]	Strongly *NP*-hard [4]
ASSIGNMENT	Strongly *NP*-hard [1]	Strongly *NP*-hard [1]
KNAPSACK	Strongly *NP*-hard [43]	Strongly *NP*-hard [4]
MIN CUT	Strongly *NP*-hard [2]	Strongly *NP*-hard [2]
MIN s—t CUT	Strongly *NP*-hard [2]	Strongly *NP*-hard [2]

NP-Hardness: Spanning Tree

Theorem

The MIN-MAX REGRET SPANNING TREE *problem with discrete uncertainty* $\mathcal{U}$ *is NP-hard, even if the graph is a grid with only two rows and* $K = |\mathcal{U}| = 2$.

NP-Hardness: Spanning Tree

Theorem

The MIN-MAX REGRET SPANNING TREE *problem with discrete uncertainty* $\mathcal{U}$ *is NP-hard, even if the graph is a grid with only two rows and* $K = |\mathcal{U}| = 2$*.*

Proof.

Reduction from PARTITION. □

Regret and Multi-Objective Optimization

Theorem

Given a minimization problem P*, at least one optimal solution for* Discrete Min-Max Regret P *is necessarily an efficient solution for each* $c \in \mathcal{U}$*.*

Regret and Multi-Objective Optimization

Theorem

Given a minimization problem P*, at least one optimal solution for* $\textsc{Discrete Min-Max Regret}$ P *is necessarily an efficient solution for each* $c \in \mathcal{U}$*.*

Proof.

If $x \in \mathcal{X}$ dominates $y \in \mathcal{X}$, then $c(x) \leq c(y)$ for each $c \in \mathcal{U}$.

Regret and Multi-Objective Optimization

Theorem

Given a minimization problem P*, at least one optimal solution for* $\textsc{Discrete Min-Max Regret P}$ *is necessarily an efficient solution for each* $c \in \mathcal{U}$.

Proof.

If $x \in \mathcal{X}$ dominates $y \in \mathcal{X}$, then $c(x) \leq c(y)$ for each $c \in \mathcal{U}$.
$\Rightarrow r(x) \leq r(y)$.

Regret and Multi-Objective Optimization

Theorem

Given a minimization problem P*, at least one optimal solution for* $\textsc{Discrete Min-Max Regret}$ P *is necessarily an efficient solution for each* $c \in \mathcal{U}$*.*

Proof.

If $x \in \mathcal{X}$ dominates $y \in \mathcal{X}$, then $c(x) \leq c(y)$ for each $c \in \mathcal{U}$.
$\Rightarrow r(x) \leq r(y)$.
Take

$$\min_{x \in \mathcal{E}} r(x).$$

□

Approximation

Theorem

There exists a K-approximation algorithm for the MIN-MAX ROBUST *problem with arbitrary $\mathcal{X} \subseteq \{0,1\}^n$ which requires only solving one instance of the classic linear optimization problem for $\mathcal{X}$.*

The proof is an exercise!

Approximation

Theorem

There exists a K-approximation algorithm for the MIN-MAX ROBUST *problem with arbitrary $\mathcal{X} \subseteq \{0,1\}^n$ which requires only solving one instance of the classic linear optimization problem for $\mathcal{X}$.*

The proof is an exercise!

Theorem

If for any instance I of a robust MIN-MAX REGRET *robust optimization problem on $\mathcal{X}$ with discrete uncertainty $\mathcal{U}$ it holds that*

1. *a lower and upper bound L and U for the optimal value can be computed in time $p(|I|)$ such that $U \leq q(|I|)L$, where p and q are two polynomials with q non-decreasing and $q(|I|) \geq 1$, and*
2. *there exists and algorithm that finds for I an optimal solution $r(|I|, U)$, where r is a non-decresing polynomial,*

then there exists an FPTAS.

Approximation

Problems	Constant		Non-constant	
	Min–max	Min–max regret	Min–max	Min–max regret
SHORTEST PATH	fptas [61]	fptas [4,3]	Not $\log^{1-\varepsilon} k$ approx. [40]	Not $\log^{1-\varepsilon} k$ approx. [40]
SPANNING TREE	fptas [4,3]	fptas [4,3]	Not $\log^{1-\varepsilon} k$ approx. [40]	Not $\log^{1-\varepsilon} k$ approx. [40]
KNAPSACK	fptas [4]	Non approx. [4]	Non approx. [4]	Non approx. [4]

Thinking Assignment: What else should be robust?

Examples of things that we would like to be robust that we have already seen:

- Cost
- Regret

Thinking Assignment: What else should be robust?

Examples of things that we would like to be robust that we have already seen:

- Cost
- Regret

What other measurable bad properties of an uncertain optimization problem would one like to be robust against?

Not a literature search exercise! Be creative!

Thank you!