What Decision to Make In a Conflict Situation under Interval Uncertainty: Efficient Algorithms for the Hurwicz Approach

Bartłomiej Jacek Kubica[1], Andrzej Pownuk[2], and Vladik Kreinovich[2]
[1]Department of Applied Informatics, Warsaw University of Life Sciences
ul. Nowoursynowska 159 02-776 Warsaw, Poland
bartlomiej.jacek.kubica@gmail.com
[2]Computational Science Program, University of Texas at El Paso
El Paso, TX 79968, USA, ampownuk@utep.edu, vladik@utep.edu

1. How Conflict Situations Are Usually Described

- In many practical situations – e.g., in security – we have conflict situations.
- For example, a terrorist group wants to attack one of our assets, while we want to defend them.
- For each possible pair of strategies (i, j), let u_{ij} be the gain of the first side (negative if this is a loss).
- Let v_{ij} be the gain of the second side.
- A *conflict situation* is when we cannot improve v without worsening u.
- *Example: zero-sum games*, when $v_{ij} = -u_{ij}$.
- While zero-sum games are a useful approximation, they are not always a perfect description of the situation.

2. Describing Conflict Situations (cont-d)

- For example, the main objective of the terrorists may be publicity, so:
  - a small attack in the country's capital may not cause much damage but bring media attention,
  - a serious attack in a remote are may be more damaging but not as media-attractive.
- To take this difference into account, we need, for each pair of strategies (i, j), to describe both:
  - the gain u_{ij} of the first side and
  - the gain v_{ij} of the second side.
- In general, we do not necessarily have $v_{ij} = -u_{ij}$.

3. It Is Often Beneficial to Act Randomly

- If we only one security person and two objects to protect, then we can:
  - post this person at the first objects and
  - post him/her at the second object.
- If we follow one of these strategies, then the adversary will attack the other (unprotected) object.
- It is thus more beneficial to assign the security person to one of the objects at random.
- This way, for each object of attack, there will be a 50% probability that this object will be defended.
- In general, the first side's strategy can be described by the probabilities $p_1, \ldots, p_n$ of selecting an arrangement:

$$\sum_{i=1}^{n} p_i = 1.$$

4. Toward Precise Formulation of the Problem

- Similarly, the second side selects probabilities $q_1, \ldots, q_m$ for which $\sum_{j=1}^{m} q_j = 1$.
- The expected gains of the two sides are:

$$g_1(p,q) = \sum_{i=1}^{n} \sum_{j=1}^{m} p_i \cdot q_j \cdot u_{ij} \text{ and } g_2(p,q) = \sum_{i=1}^{n} \sum_{j=1}^{m} p_i \cdot q_j \cdot v_{ij}.$$

- Once the 1st side selects the probabilities p_i, the 2nd side knows them – simply by observing the past history.
- So, the 2nd side selects a strategy q for which its gain is the largest possible:

$$g_2(p, q(p)) = \max_q g_2(p,q).$$

- Similarly, the 2nd side select a strategy q for which

$$g_2(p(q), q) \to \max_q, \text{ where } p(q) \stackrel{\text{def}}{=} \arg\max_p g_1(p,q).$$

5. Towards an Algorithm for Solving this Problem

- Once the strategy p is selected, the 2nd side selects q that maximizes $g_2(p, q)$.
- The expression $g_2(p, q)$ is linear in terms of q_j.
- Thus, $g_2(p, q)$ is the convex combination of gains corr. to deterministic strategies:

$$g_2(p, q) = \sum_{j=1}^{m} q_j \cdot g_{2j}(p), \text{ where } g_{2j}(p) \stackrel{\text{def}}{=} \sum_{i=1}^{n} p_i \cdot v_{ij}.$$

- So, the largest possible gain is attained when q is a deterministic strategy.
- The j-th strategy is selected it is better than others:

$$\sum_{i=1}^{n} p_i \cdot v_{ij} \geq \sum_{i=1}^{n} p_i \cdot v_{ik} \text{ for all } k \neq j.$$

6. Towards an Algorithm (cont-d)

- For strategies p for which the second side selects the j-th response, the gain of the 1st side is $\sum_{i=1}^{n} p_i \cdot u_{ij}$.
- Among all strategies p with this "j-property", we select the one with max expected gain of the 1st side.
- This can be found by optimizing a linear function under constraints which are linear inequalities.
- It is known that for such *linear programming* problems, there are efficient algorithms.
- Then, we find j for which the gain is the largest.

7. An Algorithm for Solving the Problem

- For each j from 1 to m, we solve the following linear programming (LP) problem:

$$\sum_{i=1}^{n} p_i^{(j)} \cdot u_{ij} \to \max_{p_i^{(j)}} \text{ under the constraints}$$

$$\sum_{i=1}^{n} p_i^{(j)} = 1,\ p_i^{(j)} \geq 0,\ \sum_{i=1}^{n} p_i^{(j)} \cdot v_{ij} \geq \sum_{i=1}^{n} p_i^{(j)} \cdot v_{ik} \text{ for all } k \neq j.$$

- We then select $p^{(j)} = \left(p_1^{(j)}, \ldots, p_n^{(j)}\right)$, $1 \leq j \leq m$ for which the value $\sum_{i=1}^{n} p_i^{(j)} \cdot u_{ij}$ is the largest.
- *Comment.* Solution is simpler in zero-sum situations, where we only need to solve one LP problem.

8. Need for Parallelization

- When each side has a small number of strategies, the corresponding problem is easy to solve.
- However, e.g., when we assign air marshals to different international flights, the number of strategies is huge.
- Then, the only way to solve the problem is to perform at least some computations in parallel.
- Good news: all m linear programming problems can be solved on different processors.
- Not so good news: programming problems are P-hard, i.e., provably the hardest to parallelize efficiently.

9. Need to Take Uncertainty into Account

- In practice, we rarely know the exact gains u_{ij} and v_{ij}.
- At best, we know the *bounds* on these gains, i.e., we know:
  - the interval $[\underline{u}_{ij}, \overline{u}_{ij}]$ that contains the actual (unknown) values u_{ij}, and
  - the interval $[\underline{v}_{ij}, \overline{v}_{ij}]$ that contains the actual (unknown) values v_{ij}.
- It is therefore necessary to decide what to do in such situations of interval uncertainty.

10. How Interval Uncertainty is Taken into Account Now

- In the above description of a conflict situation, we mentioned that:
  - when we select the strategy p,
  - we maximize the worst-case situation, i.e., the smallest possible gain $\min\limits_{q} g_1(p, q)$.
- It seems reasonable to apply the same idea to the case of interval uncertainty.
- So, we maximize the smallest possible gain $g_1(p, q)$:
  - over all possible strategies q of the 2nd side *and*
  - over all possible values $u_{ij} \in [\underline{u}_{ij}, \overline{u}_{ij}]$.
- Efficient algorithms for such worst-case formulation have indeed been proposed.

11. Need for a More Adequate Solution

- The adversary wants to minimize our gain, so the worst-case approach makes sense.
- For interval uncertainty, the most adequate idea is to select the alternative a that maximizes:

$$u^H(a) \stackrel{\text{def}}{=} \alpha \cdot \overline{u}(a) + (1-\alpha) \cdot \underline{u}(a).$$

- Here $\alpha \in [0,1]$ describes the decision maker's attitude.
- This expression was first proposed by Polish-American Nobelist Leonid Hurwicz.
- For $\alpha = 0$, we optimize the worst-case value $\underline{u}(a)$.
- For other values α, we have different optimization problems.

12. Analysis of the Problem

- Once both sides select strategies p and q, the gain of the 2nd side can take any value from

$$\underline{g}_2(p,q) = \sum_{i=1}^{n}\sum_{j=1}^{m} p_i \cdot q_j \cdot \underline{v}_{ij} \text{ to } \overline{g}_2(p,q) = \sum_{i=1}^{n}\sum_{j=1}^{m} p_i \cdot q_j \cdot \overline{v}_{ij}.$$

- According to Hurwicz's approach, the 2nd side selects a strategy q that maximizes

$$g_2^H(p,q) \stackrel{\text{def}}{=} \alpha_v \cdot \overline{g}_2(p,q) + (1-\alpha_v) \cdot \underline{g}_2(p,q).$$

- This expression can be represented as $g_2^H(p,q) = \sum_{i=1}^{n}\sum_{j=1}^{m} p_i \cdot q_j \cdot v_{ij}^H$, where $v_{ij}^H \stackrel{\text{def}}{=} \alpha_v \cdot \overline{v}_{ij} + (1-\alpha_v) \cdot \underline{v}_{ij}$.

- Under the above strategy $q = q(p)$ of the 2nd side, the 1st side gains the value $g_1(p, q(p)) = \sum_{i=1}^{n}\sum_{j=1}^{m} p_i \cdot q_j \cdot u_{ij}$.

13. Analysis of the Problem (cont-d)

- We do not know the exact values u_{ij}, we only know the bounds $\underline{u}_{ij} \le u_{ij} \le \overline{u}_{ij}$.
- So, all we know is that this gain will be between

$$\underline{g}_1(p, q(p)) = \sum_{i=1}^{n} \sum_{j=1}^{m} p_i \cdot q_j \cdot \underline{u}_{ij} \text{ and } \overline{g}_1(p, q(p)) = \sum_{i=1}^{n} \sum_{j=1}^{m} p_i \cdot q_j \cdot \overline{u}_{ij}.$$

- According to Hurwicz's approach, the 1st side should select a strategy p that maximizes

$$g_1^H(p, q) \stackrel{\text{def}}{=} \alpha_u \cdot \overline{g}_1(p, q(p)) + (1 - \alpha_u) \cdot \underline{g}_1(p, q(p)).$$

- This expression has the form $g_1^H(p, q) = \sum_{i=1}^{n} \sum_{j=1}^{m} p_i \cdot q_j \cdot u_{ij}^H$, where $u_{ij}^H \stackrel{\text{def}}{=} \alpha_u \cdot \overline{u}_{ij} + (1 - \alpha_u) \cdot \underline{u}_{ij}$.
- The resulting optim. problem is the same as in the no-uncertainty case, with u_{ij}^H, v_{ij}^H instead of u_{ij}, v_{ij}.

14. What Is Known: Reminder

- For every deterministic strategy i of the 1st side and for every deterministic strategy j of the 2nd side:
  - we know the interval $[\underline{u}_{ij}, \overline{u}_{ij}]$ of the possible values of the gain of the 1st side, and
  - we know the interval $[\underline{v}_{ij}, \overline{v}_{ij}]$ of the possible values of the gain of the 2nd side.
- We also know the parameters α_u and α_v characterizing decision making of each side under uncertainty.

15. Algorithm for Solving Conflict Situation under Hurwicz-Type Interval Uncertainty

- First, we compute the values

$$u_{ij}^H \stackrel{\text{def}}{=} \alpha_u \cdot \overline{u}_{ij} + (1-\alpha_u) \cdot \underline{u}_{ij} \text{ and } v_{ij}^H \stackrel{\text{def}}{=} \alpha_v \cdot \overline{v}_{ij} + (1-\alpha_v) \cdot \underline{v}_{ij}.$$

- For each j, we solve the following linear programming problem:

$$\sum_{i=1}^{n} p_i^{(j)} \cdot u_{ij}^H \to \max_{p_i^{(j)}} \text{ under the constraints}$$

$$\sum_{i=1}^{n} p_i^{(j)} = 1, \ p_i^{(j)} \geq 0, \ \sum_{i=1}^{n} p_i^{(j)} \cdot v_{ij}^H \geq \sum_{i=1}^{n} p_i^{(j)} \cdot v_{ik}^H \text{ for all } k \neq j.$$

- We select a solution $p^{(j)} = \left(p_1^{(j)}, \ldots, p_n^{(j)}\right)$ that maximizes $\sum_{i=1}^{n} p_i^{(j)} \cdot u_{ij}^H$.

16. Zero-Sum Case

- In the no-uncertainty case, zero-sum games are easier to process.
- Let us consider situations in which possible values v_{ij} are exactly values $-u_{ij}$ for possible u_{ij}:

$$[\underline{v}_{ij}, \overline{v}_{ij}] = \{-u_{ij} : \underline{u}_{ij} \in [\underline{u}_{ij}, \overline{u}_{ij}]\}.$$

- In this case, $\underline{v}_{ij} = -\overline{u}_{ij}$ and $\overline{v}_{ij} = -\underline{u}_{ij}$.
- Then, $v_{ij}^H = \alpha_v \cdot \overline{v}_{ij} + (1 - \alpha_v) \cdot \underline{v}_{ij}$ and $u_{ij}^H = \alpha_u \cdot \overline{u}_{ij} + (1 - \alpha_u) \cdot \underline{u}_{ij}$.
- One can check that the resulting game is zero-sum (i.e., $v_{ij}^H = -u_{ij}^H$) only when $\alpha_u = 1 - \alpha_v$.
- In all other cases, the general algorithm will be needed, without a zero-sum simplification.

17. Conclusion

- In this talk, we show how to take interval uncertainty into account when solving conflict situations.
- Such algorithms are known when each side of the conflict maximizes its worst-case expected gain.
- A general Hurwicz approach provides a more adequate description of decision making under uncertainty.
- In this approach, each side maximizes the convex combination of the worst-case and the best-case gains.
- We describe how to resolve conflict situations under the general Hurwicz approach to interval uncertainty.

18. Acknowledgments

This work was supported in part:

- by the National Science Foundation grants:
  - HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and
  - DUE-0926721, and
- by an award from the Prudential Foundation.