Uncertainty and Robustness Analysis, Evolutionary Computation and MCDM

Daniel Salazar\(^{(1,2)}\), Blas Galván\(^{(2)}\), Claudio Rocco\(^{(3)}\)

(1) Dept. 3MI. Ecole des Mines de Saint Etienne.
   salazar@emse.fr, danielsalazaraponte@gmail.com

(2) Instituto de Ingeniería Computacional SIANI.
   Universidad de Las Palmas de Gran Canaria.
   bgalvan@step.es, bgalvan@iusiani.ulpgc.es

(3) Facultad de Ingeniería. Universidad Central de Venezuela
   croccoucv@gmail.com
Motivation and scope (1/2)

- Many problems in engineering and related sciences are ultimately of decision-making.

- Frequently multiple-criterion formulations better model DM’s concerns → MCDM.

- Multiple Objective Evolutionary Algorithms (MOEA) became a popular as all-purpose tools for solving academic and real problems.

- Although their success, the attempts to handle uncertainty when using MOEA are still scarce.
Motivation and scope (2/2)

- Facts:
  - EMO authors usually pursue all-purpose algorithms, but they are not useful in uncertain environments.
  - Some EMO authors proposed very particular algorithms. They can be effective but only on a wide range of situations.

- We consider in this work the alternatives for solving MCDM problems under uncertainty when MOEA are used as solving tools.
Working premises

1. The words of uncertainty (*uncertainty, robustness, risk, reliability…*) don’t have unique semantic meanings. Hence:
   
   i. Criteria for coping with uncertainty depend on the DM, the problem and the state of affair.
   
   ii. Uncertainty is understood hereafter as lack of data or information.

2. The available information (*type and amount*) determines the theoretical framework(s) applicable for representing uncertainty.
Understanding MOEA

Multiple-objective problem:

Evolutionary process:

Quality assessment $F_a(F(x))$:

Optimality goal

Diversity goal
Effect of Uncertainty

**Epistemic case:**

**Uncertain function case:**

**Aleatory case:**
AUREO: Analysis of Uncertainty and Robustness in Evolutionary Optimization

- Methodological framework:
  - 1<sup>st</sup> Stage: Find a suitable mathematical program regarding the actual level of uncertainty.
    - Interest on identification of sources and characterization of uncertainty.
    - Interest on criteria.
  - 2<sup>nd</sup> Stage: Define the algorithmic structure. (new or adapted MOEA) for the resulting program.
    - Interest on implementation and efficiency.
General R-seeking program

- In the presence of uncertainty the original problem $\text{Opt } F(x)$ s.t. $G(x)$ is transformed into one of optimization of $R$ (*robustness, reliability*) s.t. $G(x)$ and $I(\cdot)$:

\[
\text{Opt} \left( R \left( F(x), x, p, \delta_x, \delta_p, \gamma \right) \right) \\
\text{s.t. } x \in X, p \in P \\
\delta_x \leq x \leq \overline{\delta}_x \\
\delta_p \leq p \leq \overline{\delta}_p \\
G(x, p, \delta_x, \delta_p) \leq 0 \\
I(F(x), x, p, \delta_x, \delta_p, \gamma) \leq 0
\]

- Quantities $\delta_x$ and $\delta_p$ serve to model the uncertainty as:

\[ [x + \delta, x + \overline{\delta}] \]

- Constraints $I(\cdot)$ are auxiliary criteria set at the actual level of information
Class 1: Uncertainty propagating programs (1/2)

- Input well defined: $\delta_x$, $\delta_p$
- Information constraints $I(\cdot)$: Possibly

- Many propagation methods:
  - Monte Carlo, Convolution, Extension principle, Interval arithmetic…

- Quality assessment based on representative quantities like
  - Mean, median, quantiles, variance, extreme values.
  - or other criteria:
    - Savage, Hurwicz, non prob heuristics.
## Class 1: Uncertainty propagating programs (2/2)

<table>
<thead>
<tr>
<th>Realm of study</th>
<th>Authors and works</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Fuzzy logic:</td>
<td></td>
</tr>
<tr>
<td>• Dempster-Shafer:</td>
<td></td>
</tr>
<tr>
<td>Computing mean and variance</td>
<td>Kreinovich et al. (2006)</td>
</tr>
<tr>
<td>• p-boxes:</td>
<td></td>
</tr>
<tr>
<td>Estimation of the expected value</td>
<td>Bruns et al. (2006)</td>
</tr>
<tr>
<td>• Intervals:</td>
<td></td>
</tr>
<tr>
<td>• Monte Carlo simulation:</td>
<td></td>
</tr>
<tr>
<td>Techniques for processing interval uncertainty in engineering</td>
<td>Kreinovich et al. (2004)</td>
</tr>
</tbody>
</table>
Class 2: domain seeking programs

- Input ill defined: $\delta_x, \delta_p$?
- Information constraints $I(\cdot)$: Mandatory

Investigate the maximal domain for $\delta_x, \delta_p$ regarding nominal vectors $x$ and $p$. While the shape of the “box” is not uniquely defined, we suggest hyper-rectangles.
Class 3: Mixed seeking procedure

- Input ill defined: $\delta_x$, $\delta_p$?
- Information constraints $I(\cdot)$: ?

- Elicitate $\delta_x$, $\delta_p$ and reduce to a Class 1, or
- Suppose $\delta_x$, $\delta_p$ and solve like a Class 1 in order to help the DM to define $I(\cdot)$, then solve as a Class 2.
Some experiences:

- We have applied the AUREO methodology to:
  - Design algorithms to solve Class 1 problems.
  - To solve Class 2 and Class 3 dependability problems using NSGA-II and MOSA.
Final remarks

- As practitioners have to deal with MCDM under uncertainty, we propose:
  - Broaden the view: adapt tools (MOEA) to problems, don’t do the opposite.
  - Split the problem: identify the class of problem your dealing with and exploit its characteristics mathematically and algorithmically.
  - Iterate: find the mathematical program, adapt your MOEA and check efficiency. If your results are poor go back to the program and refine everything from the top.
Open problems and on-going research

- To enhance and develop MOEA for Class 1 problems using non classical probability theory (fuzzy logic, possibility, imprecise prob).

- Investigating the semantic and the implementation of Class 2 using non classical probability theory (if any).
Thank you!

It's time for questions...