NoLoadj project - Phan Manh Tung - Week 1 (Report 1 - 06/03/2023)

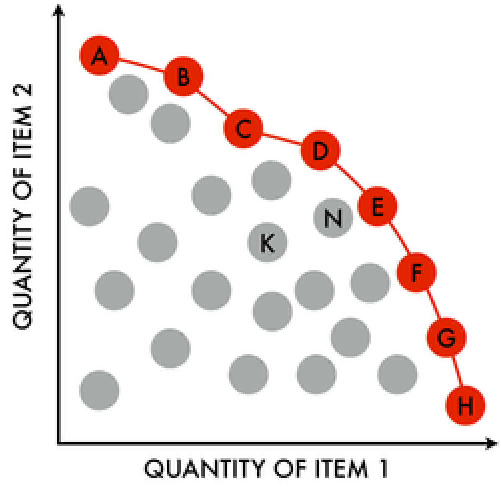
I. Problem definition

Multi-objective optimization: How to optimize multiple objectives when they are in conflict ? There is alway trade-off.
Our solution is to make a data-driven decision maker using Pareto frontier.

This solution is applicable to many industries, for instance:

+ economics
+ supply chain management
+ genome analysis
+ manufacturing

Example:



An example of Pareto Front optimizing 2 objectives.

1. ***Dominated*** (Gray points): These are not optimal because there is always another solution that has a better performance on both objectives.

2. Pareto Optimal = ***Non Dominated*** (Red points) The set of optimal solutions, which are considered equally. This set create the Pareto Front. In order to identify which solutions are better from this set, we need a ***subjective weighting***.

Prior constraint: If there is no prior constraint, we explore the full N-dimensional space. If there are prior constraints, we need to adjust the algorithm accordingly.

II. Existing solutions

1. Analysis: There are many implementations for the problem. I found 3 good examples on the Internet: 1. the naive approach (bigger is better); 2. ε-constraint with pyomo; 3. functions implementing NSGA-II algorithm.
   Link:
   https://colab.research.google.com/drive/1K2hmR1fVPp_e9emSFTQoIW0LP5M0ZBka?usp=sharing

   I will give my original thoughts in the below sections:

2. Key features
   + Overall good implementations of the algorithm, which are executed really fast and efficiently.
   + There is a good control of the input and the expected output.
   + There are good visualizations.

3. Limitations
   + There is a lack of explanation for each steps (the expected size and shape of the input, its components and objectives; meanings of the outputs and visualization..) in order to help novice users to understand to whole process.

III. Observations from my novice experience

Considering myself as novice user, I experienced some pain points when using some of the implementation, which will be some ideas to improve users' experience:

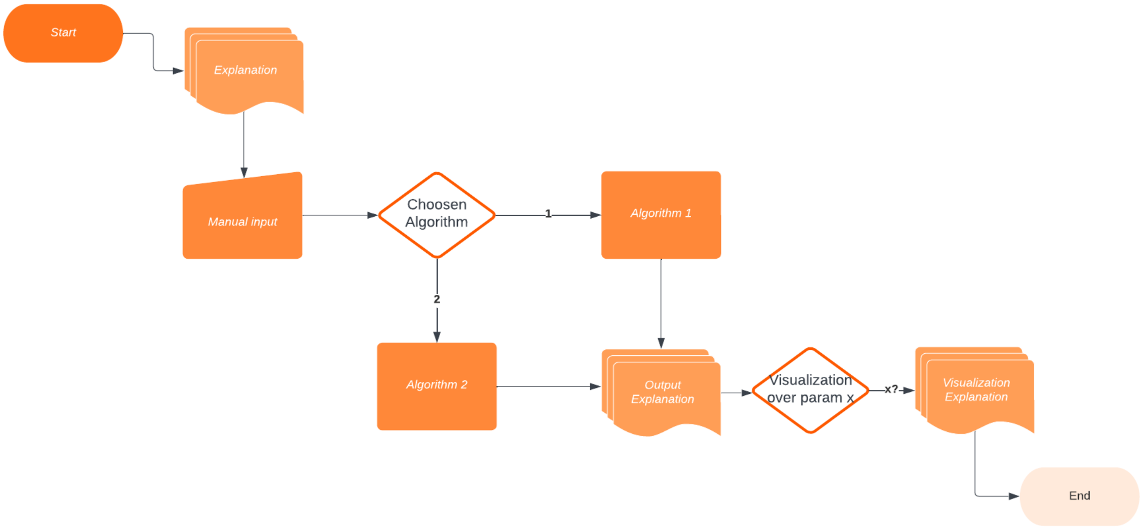
1. The multi-objective problem itself is complicated, there are many parameters to consider. Therefore, it will be better to **simplify** some technical in-between params during the process and solely

focus on input and output.

2. **More explanations** (written text or visualization) to all the necessary terms and params, which allows users having more control over the whole process.

IV. UML Suggestions

Based on the aforementioned pain points, I have a suggestion of a UML process diagram for the software as follows:



Main ideas: The digram describes a process with detailed guide for an user in each step, helping them interpret the whole process easier and therefore get more intuitions from it.

V. My expected plan

There are 2 major directions:

1. **Focus on the underlined multi-objective algorithm to improve its performance.**

2. Try to improve the users' experience by working mainly on the input-output process and visualization. Consider the algorithm as a black box.