Survey of Multi-Objective Optimization Methods for Engineering: A Study and Analysis

Written by: Brian Kurzeja, Miranda Hernandez, Nailea Ibarra, Sarida Ngo

---

Abstract - Multi-Objective Optimization is a critical area of study that allows for the understanding of different methods for solving problems where multiple conflicting objectives must be considered simultaneously. Multi-Objective Optimization offers the option to generate a set of trade-off solutions known as the Pareto Optimal Set. With these methods of analyzing solutions we can get the best possible compromises accounting for performance, cost, and efficiency. Within the Multi-Objective Optimization there are different methods- Propori, Posteriori, and No Articulation. There are different factors that must be taken into account, which will allow for the most optimized decision or choice. There is a fourth method, Genetic Algorithms, that we will analyze, and how even if it is more costly, it can yield the most "natural" results. In this analysis, the reader will learn and understand the foundations and methods of Multi-Objective Optimization Methods and will see its practical uses in optimizing applications with economic equilibrium, or even gameplay strategies.

I. Introduction

Multi-Objective Optimization (MOO) is a framework used to find optimal solutions to problems which may have multiple objectives, under constraints or not. MOO is ideal for producing a set of trade-off solutions which is known as the Pareto Optimal Set.

MOO has deep foundational uses in areas such as economic equilibrium, welfare theories, game theory, and pure mathematics. It helps to design efficient systems that typically satisfy different performance requirements, simultaneously. This can include looking at cost vs. performance or performance vs. speed, among other things.

The central concept in MOO is the Pareto Optimality which is categorized into three types:

- **Pareto Optimal:** Inability to improve an objective without worsening another.
- **Weakly Pareto Optimal:** A solution where the objective cannot be improved without worsening at least one other solution.
- **Properly Pareto Optimal:** Solution that ensures at least one objective is better off and no objective is worse off.

II. Problem Statement

One of the main challenges in MOO is defining how preferences will be expressed. The solution is the classification of MOO methods. The three categories are: priori, posteriori, and no preference articulation. The main difference, which we will see in this paper, is how and when preferences are integrated into the optimization process. In addition to these three categories, we also look at a fourth solution, genetic algorithms. The uses, purpose, benefits and limitations of these four options is what makes them each unique and adaptive challenge solutions.

III. Categories of MOO Methods

Multi-Objective Optimization methods can be classified on a number of choices. If the user has preferences to consider or no preferences at all. This decision will help to determine which method is the best option for that user and for that problem.

A. Priori Method: Decide First, Optimize Later

Preferences are set before running the optimization. The user will note what needs to be prioritized. Weights will be assigned to scale features before the algorithm gets to work and the solutions that will be given using this method will take into account the preferences of the user and only give the most useful solutions. Some common examples include:

1. Weighted Global Criterion: Objective functions are combined into a single function where weights are assigned to scale features.
2. Weighted Sum Method: Assigns weights to each objective and optimizes the combined score.
3. Lexicographic Method: Prioritizes objectives in order and optimizes one at a time.
4. Physical Programming: Instead of weights, it uses qualitative satisfaction.

B. Posteriori Method: Explore, Then Decide

With this method, a variety of optimal solutions are generated which then allows the user to decide which solutions best fit their goals based on preferences or observed trade-offs. This method is ideal when preferences are still unclear. Examples include:

1. Normal Boundary Intersection and Normal Constraint: Optimal for producing evenly spaced Pareto solutions.
2. Physical Programming: May be useful to filter the different solutions possible.

C. No Articulation of Preference: Let the Algorithm Decide

These methods do not rely on user input at all. The solutions yielded from these methods are a balance across all objectives. It also does not specify which objectives are more important before optimizing therefore preventing large bias to one objective. An example:

1. Min-Max Method: Which minimizes the worst-performing objective to find a balance between objectives.

D. Genetic Algorithms

While this algorithm may fall under the No Articulation of Preferences, genetic algorithms can be much more effective for complex or high-dimensional problems, working across all preference categories.

This method explores all possible trade-offs between multiple objectives and effectively finds a balance between exploration and exploitation to maintain diverse and refined solutions.

IV. Limitations and Challenges

Despite the strengths of Multi-Objective Optimization, there are several challenges that will limit the effectiveness of the algorithms.

A major challenge comes from computational complexity in high dimensional problems. As the objective increases, so does the complexity and the quantity of possible solutions, making exploring all possible trade-offs increasingly difficult, requiring more computational power, perhaps more than is available.

Another limitation would be choosing the best solution. As more objectives means more solutions, finding and choosing the optimal solution can be an arduous task.

Each MOO technique has its own strengths and weaknesses. Understanding is required to choose the most efficient method. For example:

- Priori methods are most useful when prioritized objectives are already known, but these methods do not allow for priorities to shift.
- Posteriori methods are ideal when exploring complex-tradeoffs, but can be computationally expensive and yield too many options for the user.
- No Articulation of Preference methods can be used when trade-offs are unknown, but may result in meaningless results.
- Genetic Algorithms allow for a dynamic exploration of solutions and optimal setups, yet they require a lot more time than other algorithms, and may not be the best option for users.

Yet, these limitations and challenges do not hinder the results of these algorithms. While there are some drawbacks, the benefits can be worth the extra understanding it takes to be able to choose the optimal solution.

V. What We Would Do Differently

If we were to present this project again, we would revise our slides and remove redundancy. In some areas, we may have over explained concepts that could have been presented more clearly and concisely without losing clarity. This feedback helped us realize the importance of focusing on the main points and finding a way to explain them in a much less complicated way, to keep the audience consistently engaged.

Additionally, it was suggested that this presentation would be more effective after our second presentation. While we understand why this might be an effective change, we believe that the presentation of Multi-Objective Optimization, first, leads to a better understanding of topics discussed later. So starting with this analysis and survey gave necessary context for the practical uses we see later on.

VI. Conclusion

Multi-Objective Optimization (MOO) is a useful and powerful framework in engineering for solving problems with conflicting goals. Through MOO, we don't just look for the best solution, we look through many possible solutions along the Pareto front, and base the decision on what the user needs. This allows for a perfect balance between goals and resources.

Through the use of a priori, posteriori, and a no preference method, the system can optimize the process and offer multiple solutions which can be tailored to result in the best solution. Additionally, genetic algorithms can be used in instances where the alternative methods are too rigid or do not offer complex enough solutions.

No single MOO method is superior to the others. It all depends on the preferences of the user, the characteristics of the problem at hand, and the computational capabilities. Understanding MOO techniques can help solve problems that would be otherwise impossible to solve without the use of any of these methods.

While the ideas can be complex, and the algorithms may be time consuming, the benefits of understanding these methods can save time and resources in the long run and may result in better, more efficient systems.

References

Marler, R. & Arora, Jasbir. (2004). Survey of Multi-Objective Optimization Methods for Engineering. Structural and Multidisciplinary Optimization. 26. 369-395. 10.1007/s00158-003-0368-6.