Department of Computer Engineering

Bu-Ali Sina University

Evolutionary Computing Course

Assignment 2: Solving the Cones Problem with a Multiobjective Evolutionary Algorithm

By:

Ali Nafisi

Course Professor:

Professor Hassan Khotanlou

Fall 2023

TABLE OF CONTENTS

1 INTRODUCTION

In the realm of engineering and optimization, tackling multi-objective problems poses unique challenges and opportunities. One such challenge is the "cones problem," an intriguing geometrical optimization dilemma characterized by its multi-objective nature and constrained design space. This problem not only demands the identification of multiple optimal solutions but also requires navigating through feasible and unfeasible regions effectively.

The cones problem presents itself with a straightforward yet captivating mathematical formulation, offering an excellent platform for exploration and experimentation within the realm of population-based Evolutionary Algorithms (EAs). This optimization conundrum revolves around determining the optimal dimensions of a cone, denoted by its base radius (r), height (h), slant height (s), volume (V), base area (B), lateral surface area (S), and total surface area (T).

The objectives are twofold: to minimize the lateral surface area (S) and the total surface area (T). These objectives, though seemingly straightforward, intertwine intricately, requiring a nuanced approach to optimization. Furthermore, the solution landscape of the cones problem transcends a single optimal design; instead, it encompasses a Pareto frontier, representing a spectrum of trade-offs between the conflicting objectives.

Navigating through this multi-objective and constrained design space necessitates the utilization of robust optimization methodologies. In this project, we propose the application of a Multiobjective population-based Evolutionary Algorithm (MOEA) as a viable solution strategy. MOEAs offer a powerful framework for exploring and discovering Pareto optimal solutions efficiently, providing insights into the trade-offs inherent in the problem domain.

This report aims to delve into the exploration of the cones problem utilizing an MOEA approach. We will delve into the intricacies of the problem formulation, discuss the implementation of the MOEA algorithm, present experimental results, and analyze the obtained Pareto optimal solutions. Through this endeavor, we seek to contribute to the broader understanding of multi-objective optimization methodologies and their applicability to real-world engineering challenges.

2 METHODOLOGY

2.1 Representation

Each individual in the population is represented by a chromosome, which is a real-valued array of length 2. The first index denotes the base radius of the cone, while the second index corresponds to its height. However, it's worth noting that due to our mutation method, which introduces variability in the genetic makeup of individuals, the final chromosome length will be extended to 4.

2.2 Objectives

Our optimization problem involves two objective functions. The first objective is to minimize the lateral surface area of the cone, as represented by Equation 1. The second objective aims to minimize the total surface area of the cone, as indicated by Equation 3. Consequently, we represent our objectives using a real-valued array of length 2, where the first index corresponds to the lateral surface area and the second index corresponds to the total surface area of the cone. This succinct representation enables us to quantify and optimize both aspects of the cone's geometry simultaneously.

$$S = \pi r \sqrt{r^2 + h^2} \tag{1}$$

$$T = \pi r^2 + S \tag{2}$$

2.3 Fitness Evaluation

During each generation, we conduct a comprehensive assessment of each individual's fitness. This evaluation involves calculating the number of individuals within the population that dominate the given individual. The lower this domination count, the higher the fitness of the individual. This approach allows us to identify and prioritize individuals that exhibit superior performance relative to their peers within the population.

2.4 Parent Selection Method; Linear Ranking Selection

Linear Ranking Selection is a technique used in evolutionary algorithms to select individuals for reproduction based on their relative fitness within the population. The individuals are ranked according to their fitness, and each is assigned a selection probability based on its rank. In this method, a linear transformation is applied to the ranks, often in the form of a linear equation. Individuals with higher fitness ranks are assigned higher selection probabilities, promoting the likelihood of their inclusion in the next generation. This ranking process allows for a balance between selecting individuals with the best fitness and maintaining diversity in the population. Linear Ranking Selection is particularly useful in scenarios where explicit fitness values may vary widely, providing a way to emphasize the importance of relative fitness without being overly sensitive to extreme values. Equation 3 shows how to calculate selection probabilities for each individual using the linear Ranking method.

$$P_{linear-rank}(i) = \frac{2-s}{\mu} + \frac{2i(s-1)}{\mu(\mu-1)} \tag{3}$$

2.5 Crossover Method; Blend Crossover

Blend crossover (BLX-α) is an evolutionary algorithm technique introduced to generate offspring from parent individuals in a manner that extends beyond the limitations of the parents' characteristics. It operates by selecting two parent individuals, x and y, and determining the range for each characteristic of the offspring based on the differences between the corresponding characteristics of the parents. If, for a particular dimension, the value of x_i is less than y_i, a difference d_i is calculated as $y_i - xi$. The range of values for the offspring's characteristic is then defined as $[x_i - \alpha d_i, x_i + \alpha d_i]$, where α represents a constant parameter that controls the extent of variation.

This approach allows for the exploration of a broader search space for generating offspring, proportional to the distances between the parents in each dimension. By considering this expanded region for offspring generation, blend crossover enhances the exploration of potential solutions in evolutionary algorithms. The width of the range for each characteristic of the offspring varies depending on the differences between the corresponding characteristics of the parents, ensuring that the variation in each dimension is proportional to the distance between the parents in that dimension. Consequently, blend crossover facilitates increased exploration and exploitation of the solution space, contributing to the effectiveness of evolutionary algorithms in solving optimization problems.

2.6 Mutation Method; Self-adaptive Mutation

In our evolutionary process, a self-adaptive mutation operator is employed on offspring generated by the crossover operator with a probability of P_m. This operator introduces variability by adding a random number sampled from a Gaussian distribution $N(0, \sigma_1)$ to the first gene of the chromosome and $N(0, \sigma_2)$ to the second gene. Notably, the term "self-adaptive" stems from the dynamic nature of the mutation process, as the values for σ_1 and σ_2 are not predetermined. Instead, these hyperparameters are appended to the chromosome, effectively extending its length from 2 to 4. Throughout the course of the algorithm iterations, the values for these hyperparameters are optimized, allowing the system to adapt and fine-tune the mutation process for improved performance.

It's important to highlight that the mutation process involves self-mutation of σ_1 with itself and σ_2 with itself before mutating the first and second genes of the chromosome. This sequence ensures that both the scale and direction of mutation are dynamically adjusted based on the evolving conditions, thereby enhancing the exploration and exploitation capabilities of the algorithm.

2.7 Survivor Selection Method

Following the generation of offspring, our approach employs the $(\mu + \lambda)$ selection strategy. This method involves identifying all non-dominated individuals and transferring them to the subsequent generation. Additionally, we iteratively select random dominated individuals and incorporate them into the next generation until the population size of the ensuing generation aligns with the desired population size. This ensures a diverse population representation in subsequent generations, promoting effective exploration and exploitation of the solution space.

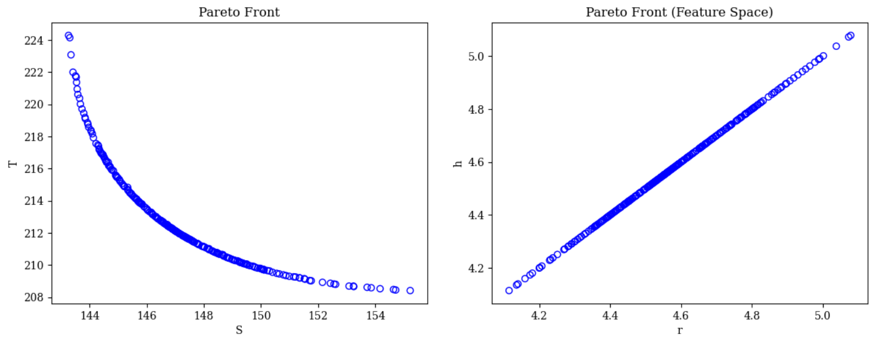

Figure 1: Pareto front.

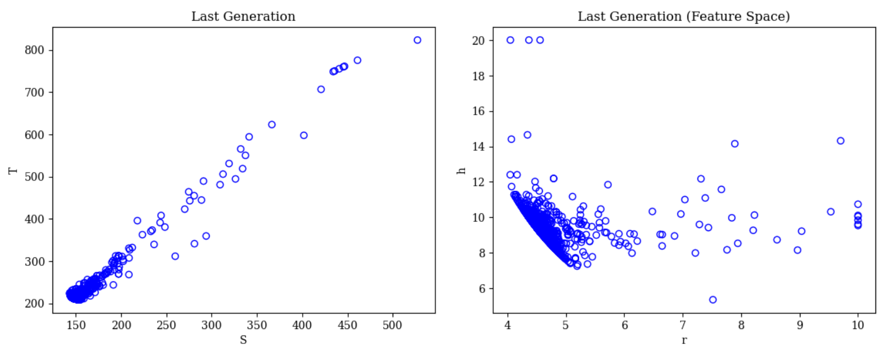

Figure 2: Last generation.

3 RESULTS

After running the algorithm for 100 iterations with a population size of 1000, a mutation probability of 0.1, and a selection parameter (s) set to 1.5 for linear selection, the outcome was exemplary. In Figure 1, the Pareto front is depicted, while Figure 2 illustrates the objective space and feature space of the final generation. Among the 1000 individuals in the last generation, 254 were found to be non-dominated.

4 CONCLUSION

The application of a multiobjective evolutionary algorithm to solve the Cones Problem has yielded promising results. Through meticulous representation of the problem space and effective fitness evaluation techniques, our methodology has successfully navigated the complexities inherent in multiobjective optimization.

Utilizing linear ranking selection for parent selection, blend crossover for crossover, and self-adaptive mutation for diversification, we ensured a robust exploration of the solution space. This comprehensive approach facilitated the discovery of high-quality solutions that span the Pareto front.

The results presented in this study, as depicted in the Pareto front and the final generation's objective and feature spaces, underscore the efficacy of our approach. With 254 non-dominated individuals identified in the final generation, we have demonstrated the algorithm's ability to produce diverse and high-performing solutions.

Moving forward, further refinements and optimizations to the algorithm could enhance its performance and applicability to a wider range of problems. Additionally, exploring the scalability of the approach and its adaptability to different problem domains would be valuable avenues for future research.

In conclusion, the multiobjective evolutionary algorithm employed in this study represents a robust and promising approach for tackling complex optimization problems such as the Cones Problem. Its ability to efficiently navigate trade-offs between competing objectives positions it as a valuable tool for decision-making in real-world scenarios.