Scalarization Techniques in Multi-Objective Optimization Problems

Francesco Natali 1945581

December 30, 2024

1 Introduction

In this report, we apply scalarization techniques to solve two multi-objective optimization problems using OPL (Optimization Programming Language). Scalarization is a method that transforms a multi-objective optimization problem into a single-objective problem by assigning weights to each objective. By adjusting these weights, we can generate a set of solutions that represent the Pareto front. The goal is to explore how different weight combinations affect the optimal solutions and to discuss the related issues.

We will analyze two problems: the first with linear objectives and the second with more complex constraints. The scalarization weights will be varied, and we will discuss the implications of these variations on the set of extreme and intermediate non-dominated points.

2 Problem 1: Scalarization with Linear Objectives

The first problem involves two decision variables x_1 and x_2, and two objective functions:

$$f_1 = -x_1 + 2x_2, \ f_2 = 2x_1 - x_2.$$

The scalarization technique is applied by introducing scalarization weights λ_1 and λ_2 such that $\lambda_1 + \lambda_2 = 1$. The scalarized objective function is:

$$\text{Maximize } \lambda_1 f_1 + \lambda_2 f_2.$$

The constraints for this problem are:

$x_1 + x_2 \leq 7,\ -x_1 + x_2 \ \ \leq 3,\ x_1 - x_2 \leq 3,\ x_1 \ \ \leq 4,\ x_2 \leq 4,\ x_1 \ \ \geq 0,\ x_2 \geq 0.$

2.1 Results for Problem 1

The following results were obtained by varying λ_1 and λ_2:

- When $\lambda_1 = 0.5$ and $\lambda_2 = 0.5$, the optimal solution is: $x_1 = 3$, $x_2 = 4$.
- When $\lambda_1 = 0.25$ and $\lambda_2 = 0.75$, the optimal solution is: $x_1 = 4$, $x_2 = 1$.
- When $\lambda_1 = 0.75$ and $\lambda_2 = 0.25$, the optimal solution is: $x_1 = 1$, $x_2 = 4$.

2.2 OPL Model for Problem 1

```
/*********************************************

OPL Model for Problem 1 Scalarization
*********************************************/
// Decision variables
dvar float+ x1;
dvar float+ x2;

// Scalarization weights
float lambda1 = 0.5; // Example value, adjust to explore the Pareto front
float lambda2 = 1 - lambda1;

// Define the scalarized objective function
maximize lambda1 * (-x1 + 2x2) + lambda2 * (2x1 - x2);

subject to {
// Constraints
x1 + x2 <= 7;
-x1 + x2 <= 3;
x1 - x2 <= 3;
x1 <= 4;
x2 <= 4;
x1 >= 0;
x2 >= 0;
}
```

```
// Display the solution
execute {
writeln("Lambda1 = ", lambda1, ", Lambda2 = ", lambda2);
writeln("Solution: x1 = ", x1, ", x2 = ", x2);
writeln("Objective 1 (f1): ", -x1 + 2x2);
writeln("Objective 2 (f2): ", 2x1 - x2);
}
```

3 Problem 2: Scalarization with Complex Constraints

The second problem involves two decision variables x_1 and x_2, with the following objective functions:

$$Z_1 = 3x_1 - 2x_2, \; Z_2 = -x_1 + 2x_2.$$

The constraints for this problem are:

$$4x_1 + 8x_2 \geq 8, \; 3x_1 - 6x_2 \leq 6, \; 4x_1 - 2x_2 \leq 14, \; x_1 \leq 6, \; -x_1 + 3x_2 \leq 15, \; -2x_1 + 4x_2 \leq 18,$$

The scalarized objective function is:

$$\text{Maximize } \lambda_1 Z_1 + \lambda_2 Z_2.$$

3.1 Results for Problem 2

The following results were obtained by varying λ_1 and λ_2:

- When $\lambda_1 = 0.5$ and $\lambda_2 = 0.5$, the optimal solution is: $x_1 = 6$, $x_2 = 5$.
- When $\lambda_1 = 0.75$ and $\lambda_2 = 0.25$, the optimal solution is: $x_1 = 6$, $x_2 = 5$.
- When $\lambda_1 = 0.25$ and $\lambda_2 = 0.75$, the optimal solution is: $x_1 = 6$, $x_2 = 7$.

3.2 OPL Model for Problem 2

```
/********************************************

OPL Model for Problem 2 Scalarization
********************************************/
// Decision variables
dvar float+ x1;
```

```
dvar float+ x2;

// Scalarization weights
float lambda1 = 0.75; // Example value, adjust to explore the Pareto front
float lambda2 = 1 - lambda1;

// Define the scalarized objective function
maximize lambda1 * (3x1 - 2x2) + lambda2 * (-x1 + 2*x2);

subject to {
// Constraints
4x1 + 8x2 >= 8;
3x1 - 6x2 <= 6;
4x1 - 2x2 <= 14;
x1 <= 6;
-x1 + 3x2 <= 15;
-2x1 + 4x2 <= 18;
-6x1 + 3*x2 <= 9;
x1 >= 0;
x2 >= 0;
}

// Display the solution
execute {
writeln("Lambda1 = ", lambda1, ", Lambda2 = ", lambda2);
writeln("Solution: x1 = ", x1, ", x2 = ", x2);
writeln("Objective 1 (Z1): ", 3x1 - 2x2);
writeln("Objective 2 (Z2): ", -x1 + 2*x2);
}
```

4 Discussion

4.1 The Number of Extreme Non-Dominated Points

The number of extreme non-dominated points depends mainly on the following factors:

- **Formulation of the objective functions:** In both problems, the number of extreme non-dominated points is influenced by the complexity of the objective functions and the constraints. The first problem, with linear objectives, tends to have fewer extreme points because the solution space is relatively simple and directly constrained by the bounds. In contrast, the second problem, with more complex constraints, might generate a larger number of extreme non-dominated points due to the increased dimensionality and variety of feasible solutions.

- **Scalarization weights:** The way in which the scalarization weights are chosen influences the number of extreme non-dominated points. For example, when $\lambda_1 = 0.5$ and $\lambda_2 = 0.5$, the solution lies somewhere in the middle of the trade-off space. As the weights shift towards one objective, the extreme points may become more distinct and separated.

4.2 Intervals in the Lambda Parameters for Which the Same Extreme Non-Dominated Points are Obtained

There are intervals in the scalarization weights (λ_1 and λ_2) for which the same extreme non-dominated points are obtained. This phenomenon occurs because:

- **Redundancy in the objective trade-offs:** If the two objectives are strongly correlated or if the weights are close to each other, variations in the lambda values may not significantly affect the optimal solution. For example, in the second problem, when $\lambda_1 = 0.75$ and $\lambda_2 = 0.25$, the result is the same as when $\lambda_1 = 0.5$ and $\lambda_2 = 0.5$, due to the similar trade-off between objectives.

- **Flat regions of the Pareto front:** If the Pareto front has flat regions, changes in the scalarization weights may not alter the solution within that range of lambda values.