Multicriteria Optimalization and Decision Analysis

Assignment 2

Matthias F. A. Aarnoutse
s2682796

Job M. C. Vink
s1869205

Niels Witte
s2685620

May 25, 2021

1 Karush Kuhn Tucker Theorem

1.1 Karush-Kuhn Tucker conditions

For the linear equasions:

$$
\begin{aligned}
f_1(x_1, x_2) = -x_1 - x_2 &\quad \rightarrow \quad \min && (1)\\
g_1(x_1, x_2) = 1 + 2x_1 - x_2 &\quad \geq \quad 0 && (2)\\
g_2(x_1, x_2) = 1 - 2x_1 - x_2 &\quad \geq \quad 0 && (3)
\end{aligned}
$$

The Karush-Kuhn Tucker conditions are:

$$
\begin{aligned}
\lambda_1 \nabla f_1(\mathbf{x}^*) + \lambda_2 \nabla g_1(\mathbf{x}^*) + \lambda_3 \nabla g_2(\mathbf{x}^*) &= \mathbf{0} && (4)\\
\lambda_2 g_1(\mathbf{x}^*) &= 0 && (5)\\
\lambda_3 g_2(\mathbf{x}^*) &= 0 && (6)
\end{aligned}
$$

1.2 Multiobjective

If we would add the objective function $f_2(x_1, x_2) = x_1 - 4x_2 \rightarrow \min$. We would have to add a condition for $\mathbf{x}^*$ to be a locally efficient point.

$$
\begin{aligned}
\lambda \prec \mathbf{0}, v \prec \mathbf{0} & && (7)\\
\lambda_1 \nabla f_1(\mathbf{x}^*) + v_1 \nabla g_1(\mathbf{x}^*) + v_2 \nabla g_2(\mathbf{x}^*) &= \mathbf{0} && (8)\\
v_1 g_1(\mathbf{x}^*) &= 0 && (9)\\
v_2 g_2(\mathbf{x}^*) &= 0 && (10)
\end{aligned}
$$

By solving this problem graphically we find the optimal solution at $x_1 = 0, x_2 = 1$. We visualised the objective function in Figure 1

2 Lagrange Multiplier and ϵ-Constraint method

2.1 Lagrange Multiplier Rule

Given the following functions relating to a triangular prism based on a equilateral triangle:

$$
\begin{aligned}
f_1(s, h) = &\; h\frac{\sqrt{3}}{4}s^2 \rightarrow max\\
f_2(s, h) = &\, 2\frac{\sqrt{3}}{4}s^2 + 3hs \rightarrow min
\end{aligned} \quad (11)
$$

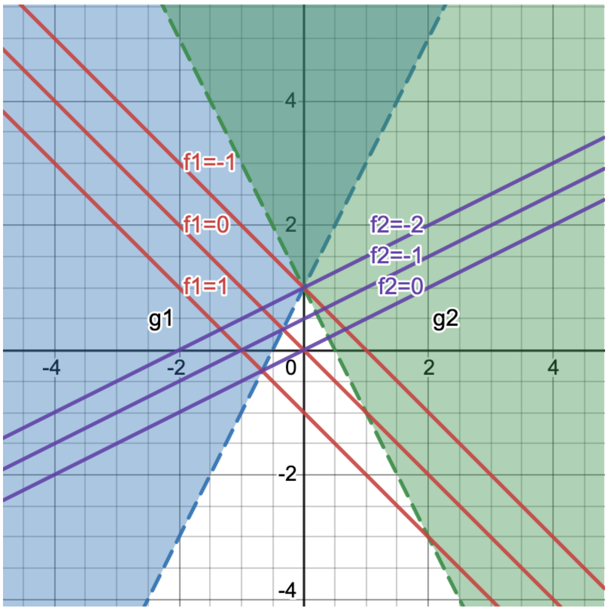

Figure 1: *The graphic used for solving the problem.*

Where f_1 is the volume which to be maximized and f_2 the total surface area which is to be minimised. We can reformulate this question using the ϵ-constraint method like this:

$$\begin{aligned} f_1(s,h) =& \quad h\frac{\sqrt{3}}{4}s^2 \rightarrow max \\ g_1(s,h) =& 2\frac{\sqrt{3}}{4}s^2 + 3hs - \epsilon \leq 0 \end{aligned} \tag{12}$$

We define the Lagrange multiplier rule as:

$$\lambda_1 \nabla f(x^*) + \sum_{i=1}^{m} \lambda_{i+1} \nabla g_i(x^*) = 0 \tag{13}$$

which expands to:

$$\begin{aligned} \lambda_1 \frac{\partial f_1}{\partial h} + \lambda_2 \frac{\partial g_1}{\partial h} = 0 \\ \lambda_1 \frac{\partial f_1}{\partial s} + \lambda_2 \frac{\partial g_1}{\partial s} = 0 \end{aligned} \tag{14}$$

We can then compute these partial derivatives to get the the following equations:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial h} &= \lambda_1(\frac{\sqrt{3}}{4}s^2) + \lambda_2(\sqrt{3}s + 3h) \\ &= \frac{\sqrt{3}}{4}\lambda_1 s^2 + \sqrt{3}\lambda_2 s + 3\lambda_2 h \\ \frac{\partial \mathcal{L}}{\partial s} &= \lambda_1(2\frac{\sqrt{3}}{4}sh) + \lambda_2(3s) \\ &= 2\frac{\sqrt{3}}{4}\lambda_1 sh + 3\lambda_2 s \end{aligned} \tag{15}$$

2.2 Determining the efficient Set

According to the slides it is worth trying $\lambda_1 = 0$ and $\lambda_1 = 1$ which results in the following equations:

From f_2 we can derive that:

$$\begin{aligned} \sqrt{3}s + 3h = & \quad 0 \\ \sqrt{3}s = & \quad -3h \\ s = & -\sqrt{3}h \end{aligned} \tag{16}$$

and

$$\begin{aligned} 3s = & 0 \\ s = & 0 \end{aligned} \tag{17}$$

this results in an infeasible solution.

3 Heuristic Multiobjective Optimization

This task states the problem of finding the maximum number of cooking in a minimum amount of dough and approximating this Pareto front.

The objectives are displayed in Equation 18 and Equation 19 where $(a,\, b)$ are the dimensions of the dough, $(x_i,\, y_i)$ the position of a cookie and s_i whether the cookie is active or not. As stated in the task description the maximum is $k = 45$ cookies based on the constraint on the dough where $a \in [10cm, 100cm]$ and $b \in [10cm, 35cm]$. This results in the aforementioned $k = \frac{35 \cdot 100}{25\pi}$ as a cookie has a diameter of $10cm$.

$$f_1(a, b, x_1, ..., x_k, y_1, ..., y_k, s_1, ..., s_k) = \sum_{i=1}^{k} s_i \tag{18}$$

$$f_2(a, b, x_1, ..., x_k, y_1, ..., y_k, s_1, ..., s_k) = ab \tag{19}$$

Making the implementation of a heuristics a tad bit easier we inverted the first objective to make it a minimization problem. This results in Equation 20. As the maximum number of cookies is 45, subtracting the number of active cookies results in our new minimization objective function.

$$f_1(a, b, x_1, ..., x_k, y_1, ..., y_k, s_1, ..., s_k) = 45 - \sum_{i=1}^{k} s_i \tag{20}$$

We played around with various heuristics and parameters. We tried 3 different selection methods. We also ended up trying (μ, λ)-selection and $(\mu + \lambda)$-selection and a discrete crossover operator. None of these however seemed to lead to any feasible results.

SMS-EMOA which is $(\mu + 1)$-selection method where if any infeasible solutions exist, a random one is deleted and a new solution is generated. If all solutions are feasible the worst one is deleted according to the crowding distance measure. This is the first method we tried.

The second selection method is tournament selection. For this method random pairs of individuals are selected, the objective values are then compared and the one with the highest value advances on to the next generation.

The third and final selection method is proportional selection. This method is also known as roulette wheel selection. Every individual has a chance to be selected based on their objective function value with individuals with a lower value having a higher chance of being selected.

We took the approach of penalizing overlapping cookies based on the distance they overlap. Our theory was that slightly overlapping cookies could mutate into feasible solutions. This however turned out to be a much bigger challenge than initially thought and thus we still have no feasible solutions after 1000 generations.

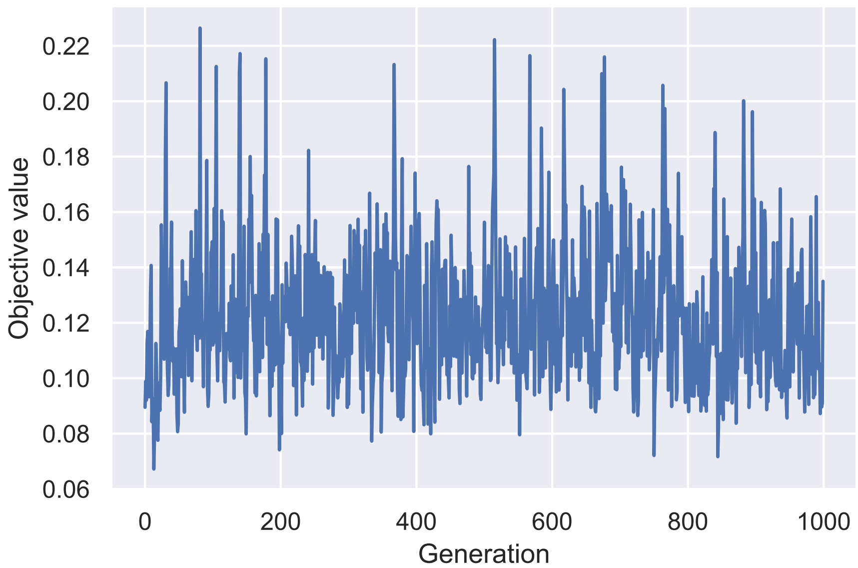

Figure 2: *Objective function value per over the course of generations. For this graph we used proportional selection*

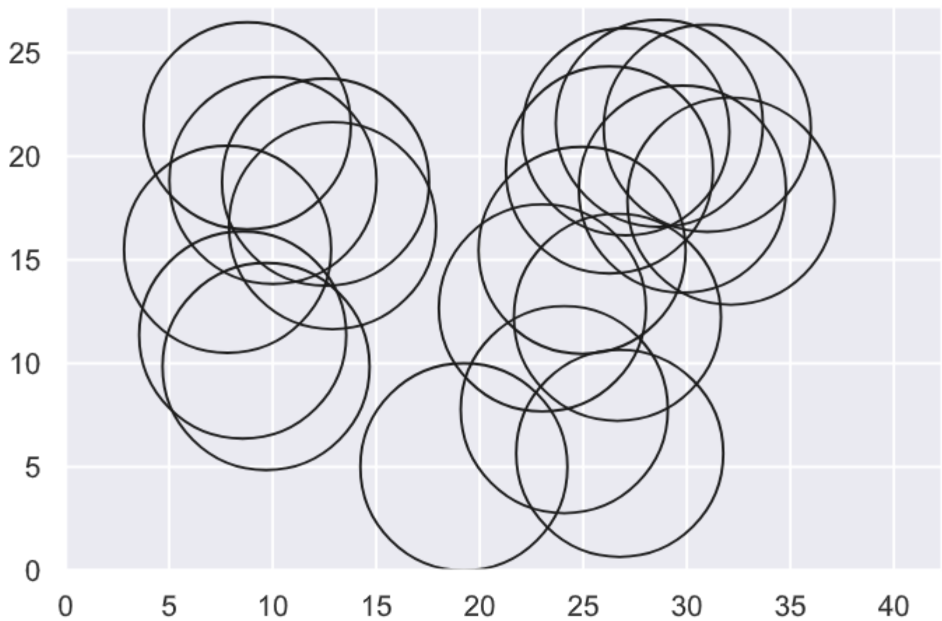

Figure 3: *The best solution after 1000 generations with proportional selection. This solution is infeasible.*

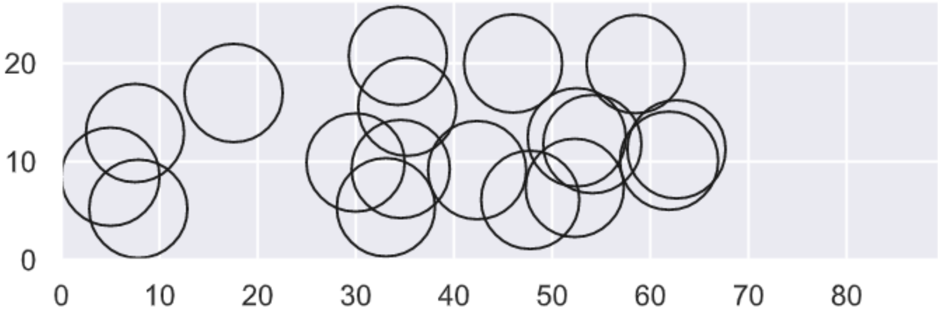

Figure 4: *The best solution after 1000 generations with Tournament selection. This solution is infeasible.*

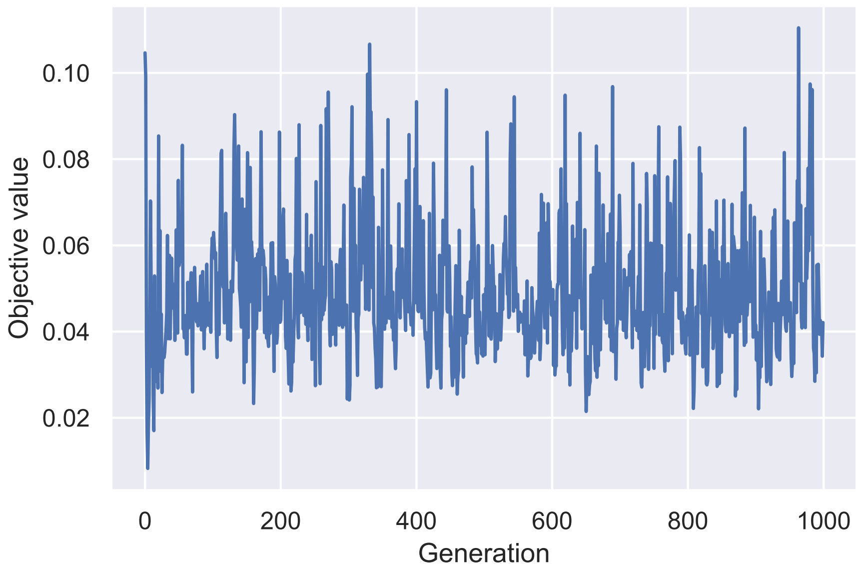

Figure 5: *Objective function value per over the course of generations. For this graph we used tournament selection*

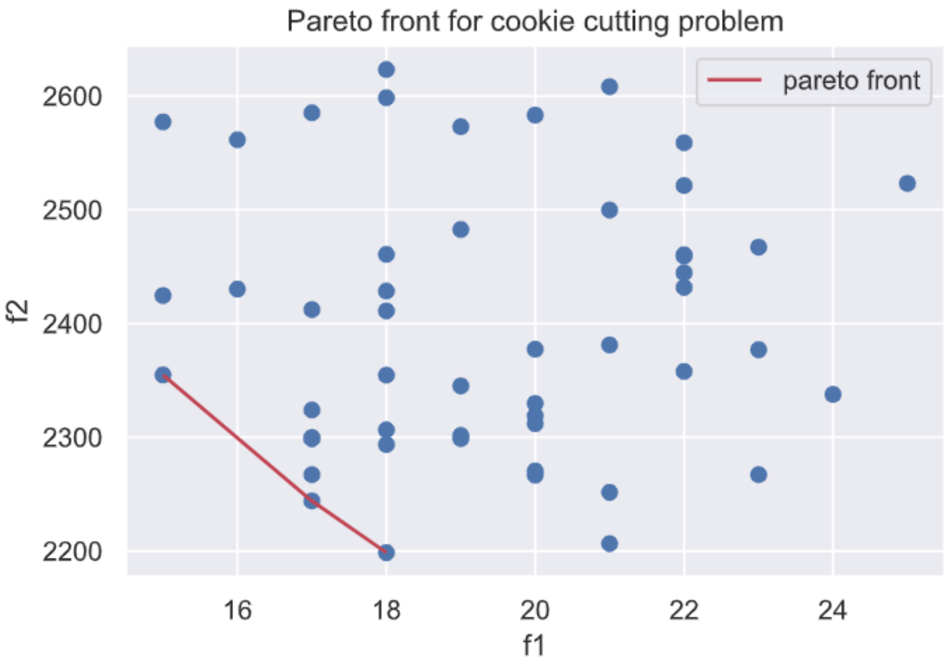

Figure 6: *Pareto front for our population after 1000 generations. f1 being the amount of cookies and f2 being the total area.*