Stochastic Optimization Algorithms 2024

Home Problems, Set 1

John Tember

Civic Registration Number: 19980731-5255

September 24, 2024

Contents

1 Problem 1.1

The task is to use the penalty method to find the minimum of the function

$$f(x_1, x_2) = (x_1 - 1)^2 + 2(x_2 - 2)^2$$

subject to the constraint

$$g(x_1, x_2) = x_1^2 + x_2^2 - 1 \leq 0$$

1.1 Define the external penalty function $f_p(\mathbf{x}; \mu)$

First, we define the external penalty function $f_p(\mathbf{x}; \mu)$, which is the sum of the objective function $f(x_1, x_2)$ and a penalty term involving μ, the penalty parameter. The function is given by:

$$f_p(x_1, x_2, \mu) = (x_1 - 1)^2 + 2(x_2 - 2)^2 + \mu \max\left(0, (x_1^2 + x_2^2 - 1)\right)^2$$

1.2 Compute the gradient $\nabla f_p(\mathbf{x}; \mu)$

Next, we compute the gradient of the external penalty function $\nabla f_p(\mathbf{x}; \mu)$. We consider two cases: when the constraint is satisfied and when it is violated.

1.2.1 Constraint is satisfied: $g(x_1, x_2) \leq 0$

When the constraint is satisfied, the penalty term does not contribute. Therefore, the gradient is simply the gradient of the objective function:

$$\nabla f_p(\mathbf{x}; \mu) = \nabla f(x_1, x_2) = \begin{pmatrix} 2(x_1 - 1) \\ 4(x_2 - 2) \end{pmatrix}$$

1.2.2 Constraint is violated: $g(x_1, x_2) > 0$

When the constraint is violated, the penalty term contributes and must be included in the gradient calculation:

$$\nabla f_p(\mathbf{x}; \mu) = \nabla f(x_1, x_2) + \nabla \left(\mu \left(g(x_1, x_2)\right)^2\right)$$

We compute the gradient of the penalty term as follows:

$$\nabla \left(\mu \left(x_1^2 + x_2^2 - 1\right)^2\right) = 4\mu \left(x_1^2 + x_2^2 - 1\right) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

Thus, the gradient in this case becomes:

$$\nabla f_p(x_1, x_2, \mu) = \begin{pmatrix} 2(x_1 - 1) \\ 4(x_2 - 2) \end{pmatrix} + 4\mu \left(x_1^2 + x_2^2 - 1\right) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

1.3 Unconstrained Minimum for $\mu = 0$

To find the unconstrained minimum, we set $\mu = 0$, which removes the penalty term. We then solve $\nabla f(x_1, x_2) = 0$. The gradient is given by:

$$\nabla f(x_1, x_2) = \begin{pmatrix} 2(x_1 - 1) \\ 4(x_2 - 2) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Solving for x_1 and x_2:

$$\begin{aligned} 2(x_1 - 1) = 0 \quad &\Rightarrow \quad x_1 = 1 \\ 4(x_2 - 2) = 0 \quad &\Rightarrow \quad x_2 = 2 \end{aligned}$$

Thus, the unconstrained minimum occurs at $(x_1, x_2) = (1, 2)$. Evaluating the function $f(x_1, x_2)$ at this point:

$$f(1, 2) = (1 - 1)^2 + 2(2 - 2)^2 = 0$$

Hence, the unconstrained minimum is located at $(1, 2)$, with a function value of $f(1, 2) = 0$.

1.4 Step Length and Penalty Parameters

We selected the following values for the step length η and the sequence of penalty parameters μ:

- Step length: $\eta = 0.0001$
- Convergence tolerance: $T = 10^{-6}$
- Sequence of penalty parameters: $\mu = 1, 10, 100, 1000$

1.4.1 Results for Different μ Values

The table below shows the values of x_1 and x_2 for different values of the penalty parameter μ, rounded to four decimal places:

μ	x_1	x_2
1	0.4338	1.2102
10	0.3314	0.9955
100	0.3137	0.9553
1000	0.3118	0.9507

Table 1: Values of x_1 and x_2 for different μ values

1.4.2 Convergence Analysis

To check for convergence, we plotted the values of x_1 and x_2 as functions of μ. The plot shows that as μ increases, both x_1 and x_2 approach stable values, indicating convergence.

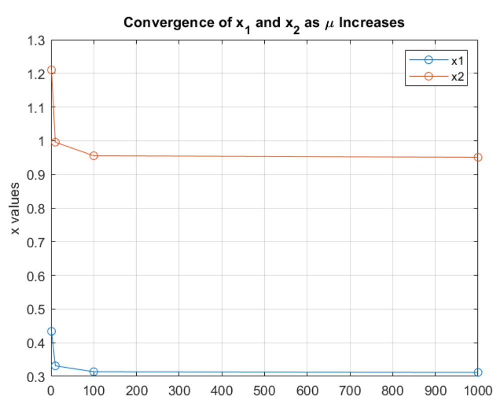

Figure 1: Convergence of x_1 and x_2 as μ increases

2 Problem 1.2: Constrained Optimization

In this problem, we aim to determine the maximum and minimum values of the function

$$f(x_1, x_2) = 4x_1^2 + 2x_2^3$$

over the closed set S, defined by

$$x_1^2 + x_2^2 \leq 4.$$

To solve this problem, we will first find critical points in the interior of S and then use the Lagrange multiplier method to analyze the boundary.

2.1 Interior Critical Points

To find the critical points in the interior of S, we begin by calculating the gradient of $f(x_1, x_2)$:

$$\nabla f(x_1, x_2) = \left(8x_1, 6x_2^2\right).$$

Setting the gradient equal to zero:

$$8x_1 = 0 \quad \text{and} \quad 6x_2^2 = 0,$$

we obtain the critical point $(x_1, x_2) = (0, 0)$ in the interior of S.

2.2 Boundary Critical Points Using Lagrange Multipliers

Next, we examine the boundary of S, defined by the equation $x_1^2 + x_2^2 = 4$, using the Lagrange multiplier method. The Lagrange function is given by:

$$\mathcal{L}(x_1, x_2, \lambda) = 4x_1^2 + 2x_2^3 + \lambda(x_1^2 + x_2^2 - 4).$$

We now compute the partial derivatives of the Lagrange function:

$$\nabla \mathcal{L}(x_1, x_2, \lambda) = (8x_1 + 2\lambda x_1, 6x_2^2 + 2\lambda x_2, x_1^2 + x_2^2 - 4).$$

Setting these partial derivatives equal to zero, we obtain the system of equations:

$$8x_1 + 2\lambda x_1 = 2x_1(4 + \lambda) = 0, \tag{1}$$

$$6x_2^2 + 2\lambda x_2 = 2x_2(3x_2 + \lambda) = 0, \tag{2}$$

$$x_1^2 + x_2^2 - 4 = 0, \tag{3}$$

Equation 1 indicates critical points when $x_1 = 0$ or $\lambda = -4$, while Equation 2 gives critical points when $x_2 = 0$ or $\lambda = -3x_2$. Therefore, we have several cases to analyze.

2.2.1 Case 1: $x_1 = 0$

Substituting $x_1 = 0$ into Equation 3, we find:

$$0^2 + x_2^2 - 4 = 0 \quad \Rightarrow \quad x_2 = \pm 2.$$

Thus, we find two boundary points at $(0, 2)$ and $(0, -2)$.

2.2.2 Case 2: $\lambda = -4$

Substituting $\lambda = -4$ into Equation 2, we find:

$$2x_2(3x_2 - 4) = 0 \quad \Rightarrow \quad x_2 = 0 \quad \text{or} \quad x_2 = \frac{4}{3}.$$

$x_2 = 0$ gives points already considered, but $x_2 = \frac{4}{3}$ provides another point of interest.

2.2.3 Case 3: $x_2 = 0$

Substituting $x_2 = 0$ into Equation 3, we find:

$$x_1^2 + 0^2 - 4 = 0 \quad \Rightarrow \quad x_1 = \pm 2.$$

Thus, we find two boundary points at $(2, 0)$ and $(-2, 0)$.

2.2.4 Case 4: $\lambda = -3x_2$

Substituting $\lambda = -3x_2$ into Equation 1:

$$2x_1(4 - 3x_2) = 0.$$

This equation doesn't give us new information beyond what we already found in previous cases.

2.2.5 Case 5: $x_2 = \frac{4}{3}$

Substituting $x_2 = \frac{4}{3}$ into the constraint Equation 3:

$$x_1^2 + \left(\frac{4}{3}\right)^2 - 4 = 0 \quad \Rightarrow \quad x_1^2 + \frac{16}{9} = 4 \quad \Rightarrow \quad x_1^2 = \frac{20}{9} \quad \Rightarrow \quad x_1 = \pm\frac{2\sqrt{5}}{3}.$$

Thus, we find two boundary points at $\left(\frac{2\sqrt{5}}{3}, \frac{4}{3}\right)$ and $\left(-\frac{2\sqrt{5}}{3}, \frac{4}{3}\right)$.

2.2.6 Summary of Critical Points

From the analysis above, the boundary critical points are:

$$(0,2), (0,-2), (2,0), (-2,0), \left(\frac{2\sqrt{5}}{3}, \frac{4}{3}\right), \left(-\frac{2\sqrt{5}}{3}, \frac{4}{3}\right).$$

We also have the interior point $(0,0)$. Now we evaluate the function $f(x_1, x_2) = 4x_1^2 + 2x_2^3$ at each of these points:

$$\begin{gathered} f(0,0) = 0, \\ f(0,2) = 16, \quad f(0,-2) = -16, \\ f(2,0) = 16, \quad f(-2,0) = 16, \\ f\left(\frac{2\sqrt{5}}{3}, \frac{4}{3}\right) = 13.63, \quad f\left(-\frac{2\sqrt{5}}{3}, \frac{4}{3}\right) = 13.63. \end{gathered}$$

2.3 Summary of Results

The results show that the maximum value of $f(x_1, x_2)$ is 16, which occurs at $(0,2)$, $(2,0)$, and $(-2,0)$. The minimum value is -16, which occurs at $(0,-2)$.

3 Problem 1.3

In this problem, we implemented a genetic algorithm (GA) for finding the minimum of the function

$$g(x_1, x_2) = (1.5 - x_1 + x_1x_2)^2 + \left(2.25 - x_1 + x_1x_2^2\right)^2 + \left(2.625 - x_1 + x_1x_2^3\right)^2$$

3.1 Parameter Experimentation and Results

After completing the program, I experimented with various values of the parameters that I was allowed to change. The parameters I choose were:

- `tournamentSize` $= 2$
- `tournamentProbability` $= 0.75$
- `crossoverProbability` $= 0.8$
- `mutationProbability` $= 0.02$
- `numberOfGenerations` $= 500$

3.1.1 Parameter Selection

To select suitable parameters, I conducted a simple parameter search where I fixed all but one parameter and varied that single parameter in each trial. Although this approach doesn't explore the entire parameter space, it provides insights into how individual parameters impact the algorithm's ability to find the best solution.

I first focused on analyzing how the parameters influenced the computational complexity. While many different combinations might yield good results, not all will do so efficiently. For example, minimizing the `numberOfGenerations` is desirable, as this directly affects the amount of computation required to find a solution.

For each parameter, the search was conducted over 100 iterations, with the algorithm required to find the optimal result within an error tolerance of 10^{-6}.

- `numberOfGenerations` was originally set to 2000, which is a relatively high value. During the parameter search, I incrementally increased the number of generations by 100 (starting at 100) in each trial. Through this process, I found that 500 generations were sufficient to meet the error tolerance.

- `tournamentSize` was already at the lowest value (2), and increasing this value would likely slow down convergence. I did, however, conduct a parameter search and found that increasing this value only allowed the algorithm to converge to a suboptimal solution. I believe this happened because increasing `tournamentSize` limits exploration by increasing the chance of always selecting the strongest individual, leading to premature convergence.

- `mutationProbability` was kept at 0.02. During the parameter search for this value, I found that only values of 0.02 and 0.03 met the requirement of finding the optimal solution. All other values failed to pass the test. As 0.02 was corresponding to $\frac{1}{\texttt{numberOfGenes}}$ (where `numberOfGenes` = 50 was of the parameters I was not allowed to change).

- `tournamentProbability` was left at 0.75 because this value strikes a good balance between exploration and exploitation. Increasing it further would make the algorithm overly exploitative, potentially leading to premature convergence, while lowering it could slow down the selection of fitter individuals. Since 0.75 is the middle ground between the lowest value (0.5) and the highest (1), I felt it was a perfect balance.

- `crossoverProbability` was kept at 0.8 as this value, along with 1, was the only ones that passed the parameter search. I chose 0.8 because it feels more intuitive, as a value of 1 would mean that the fittest individual always wins.

3.1.2 Results from 10 Runs

I ran the algorithm 10 times using the selected parameter set and obtained the following results for the maximumFitness, x_1, x_2, and the function value $g(x_1, x_2)$ (all values are rounded to 4 decimal places):

$maximumFitness$	x_1	x_2	$g(x_1, x_2)$
1.000	3.000	0.500	0.000
1.000	3.000	0.500	0.000
1.000	3.000	0.500	0.000
1.000	3.000	0.500	0.000
1.000	3.000	0.500	0.000
1.000	3.000	0.500	0.000
1.000	3.000	0.500	0.000
1.000	3.000	0.500	0.000
1.000	3.000	0.500	0.000
1.000	3.000	0.500	0.000

Table 2: Median Fitness Values for Various Mutation Rates

The results demonstrate that the algorithm reliably finds values for x_1 and x_2 that satisfy the problem's requirements, with a fitness value of approximately 1 in all runs.

3.2 Mutation Rate Analysis

The goal of this part of the problem was to familiarize ourselves with Genetic Algorithms (GAs) and explore the effect of the mutation rate (p_{mut}) on the algorithm's performance.

3.2.1 Mutation Rate Parameter Search

I extended the `RunBatch.m` file to perform several batch runs with different values of p_{mut}, ranging from 0 to 1. A total of 101 different mutation rates were explored in this analysis. However, for conciseness, only 12 representative values from the range [0, 1] are tabulated below (the rest is illustrated in the following section). These include the critical mutation rates $p_{\text{mut}} = 0$, $p_{\text{mut}} = 0.02$, and others spread across the interval.

Mutation Rate	**Median Fitness Value**
0.0	0.9942
0.02	1.0000
0.1	1.0000
0.2	0.9996
0.3	0.9992
0.4	0.9986
0.5	0.9986
0.6	0.9985
0.7	0.9986
0.8	0.9986
0.9	0.9996
1.0	0.9923

Table 3: Median Fitness Values for Various Mutation Rates

3.2.2 Plot of the Mutation Rate vs. Median Performance

In addition to the tabulated values, the full dataset of 101 mutation rates was used to create a plot, showing the relationship between the mutation rate and the median fitness value over 100 runs for each p_{mut}. The plot below clearly illustrates the behavior of the GA as the mutation rate changes.

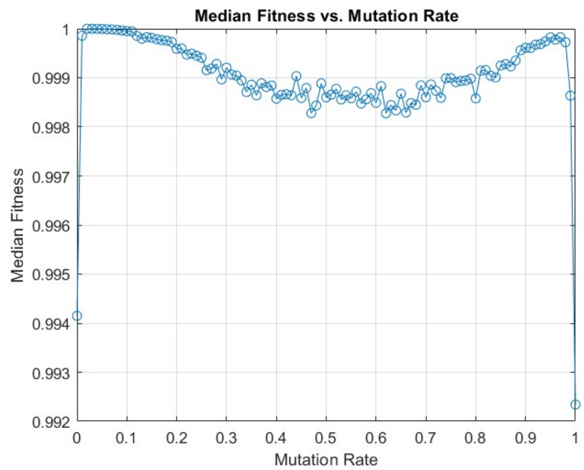

Figure 2: Convergence of x_1 and x_2 as p_{mut} increases

3.2.3 Educated Guess of Actual Minimum

It is very clear to me that the minimum is at $x_1 = 3$ and $x_2 = 0.5$. I also verified this with a analytically. As many of my results gave a fitness of 1 (or very close to it), the outputted values of x_1 and x_2 were quite accurate, which was confirmed in my analysis. The analytical solution is illustrated below:

$$g(3, 0.5) = (1.5 - 3 + 3 * 0.5)^2 + \left(2.25 - 3 + 3 * 0.5^2\right)^2 + \left(2.625 - 3 + 3 * 0.5^3\right)^2$$

$$= (1.5 - 3 + 3 * 0.5)^2 + \left(2.25 - 3 + 3 * 0.5^2\right)^2 + \left(2.625 - 3 + 3 * 0.5^3\right)^2$$

$$= (1.5 - 3 + 1.5)^2 + (2.25 - 3 + 3 * 0.25)^2 + (2.625 - 3 + 3 * 0.125)^2$$

$$= (0)^2 + (2.25 - 3 + 0.75)^2 + (2.625 - 3 + 0.375)^2$$

$$= (0)^2 + (0)^2 = 0$$