# Home Problem 1

Jesper Olsson
19990907-8991

Stochastic Optimization Algorithms, FFR105
Date: September 20, 2022

# Problem 1.1

The goal of the problem is to find the minimum of the function

$$f(x_1,x_2) = (x_1-1)^2 + 2(x_2-2)^2, \tag{1}$$

subject to the constraint

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

## 1)

We wish to find $f_p(\mathbf{x},\mu) = f(\mathbf{x}) + p(\mathbf{x};\mu)$, where $p(\mathbf{x};\mu) = \mu\left(\sum_{i=1}^{m}(\max\{g_i(\mathbf{x},0\})^2 + \sum_{j=1}^{k}\left(h_i(\mathbf{x})^2\right)\right)$. In our case, we have no equality constraints $h$, and only one inequality constraint $g$, meaning

$$p(\mathbf{x};\mu) = \mu\,(\max\{g(\mathbf{x}),0\})^2 \tag{3}$$

Thus, we define $f_p$ as

$$f_p(\mathbf{x};\mu) = \begin{cases} (x_1-1)^2 + 2(x_2-2)^2 + \mu\left(x_1^2 + x_2^2 - 1\right)^2, & if x_1^2 + x_2^2 - 1 \geq 0 \\ (x_1-1)^2 + 2(x_2-2)^2, & \text{otherwise.} \end{cases} \tag{4}$$

## 2)

For the above function $f_p(x_1,x_2;\mu)$, we calculate the gradient as $\nabla f_p(x_1,x_2;\mu) = \left[\frac{\partial f_p}{\partial x_1}, \frac{\partial f_P}{\partial x_2}\right]^T$.

In our case, we divide the gradient into two versions, depending on if the constraint $g$ is fulfilled or not. For the case where $g$ is fulfilled, i.e. in the constrained case where if $x_1^2 + x_2^2 - 1 \leq 0$, we obtain

$$\nabla f_p = \begin{bmatrix} 2x_1 - 2 \\ 4x_2 - 8 \end{bmatrix}, \tag{5}$$

where in the unconstrained case

$$\nabla f_p = \begin{bmatrix} 2x_1 - 2 + 4\mu x_1(x_1^2 + x_2^2 - 1) \\ 4x_2 - 8 + 4\mu x_2(x_1^2 + x_2^2 - 1) \end{bmatrix}. \tag{6}$$

We observe that in the unconstrained case, the gradient equals the sum of the gradient in the constrained case and $\nabla\left(\mu(x_1^2 + x_2^2 - 1)^2\right)$.

## 3)

To find a starting point for the unconstrained minimum, where $\mu = 0$, we set $\nabla f_p(\mathbf{x};\mu) = 0$, and solve for $\mathbf{x}$ to find

$$\begin{cases} 2(x_1-1) = 0 \\ 4(x_1-2) = 0 \end{cases} \implies \begin{cases} x_1 = 1 \\ x_2 = 2 \end{cases} \tag{7}$$

*Table 1: Parameter values for the penalty method runs.*

| $Parameter$ | $Value(s)$ |
|---|---|
| $\mu$ | $1, 10, 50, 100, 1000$ |
| $\eta$ | $0.0001$ |
| $T$ | $10^{-5}$ |

## Program Results

When running the program to minimize $f$, the following parameters were used:

The program gave the following result regarding the estimated values of $\mathbf{x}^* for each value of \mu$:

*Table 2: values of $x^* for different values of \mu$.*

| $\mu$ | $\mathbf{x}^* = (x_1^*, x_2^*)$ |
|---|---|
| 1 | $(0.4338, 1.2102)$ |
| 10 | $(0.3314, 0.9955)$ |
| 100 | $(0.3137, 0.9553)$ |
| 500 | $(0.3120, 0.9512)$ |
| 1000 | $(0.3118, 0.9507)$ |

We find that the values converge around ca. $(x_1^*, x_2^*) = (0.31, 0.95)$ at around $\mu = 100$. We can also visualize the convergence by plotting the values of $x_1^*, x_2^*$ by observing figure 1.

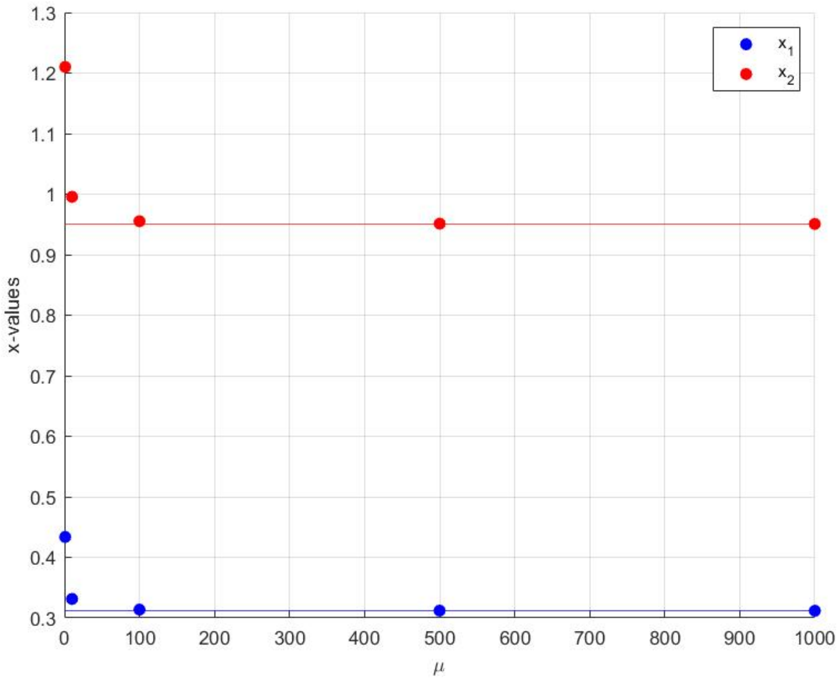


*Figure 1: Shows the obtained values of $x_1^*$ and $x_2^*$ for each value of $\mu$.*

# Problem 1.2

-

# Problem 1.3

## a)

Below, in Table 3, the chosen values of the parameters are presented.

*Table 3: Parameter values for the single runs of the GA algorithm.*

| *Parameter* | *Value* |
|---|---|
| Tournament size | 3 |
| $p_{\text{tour}}$ | 0.8 |
| $p_{\text{cross}}$ | 0.8 |
| $p_{\text{mut}}$ | 0.03 |
| # generations | 1500 |

The program was run 10 times using the above parameter values, yielding the following values for $x_1^*$, $x_2^*$, $g(x_1^*, x_2^*)$ presented in Table **??** below.

*Table 4: Results from 10 runs of the GA algorithm.*

| *Run* | $x_1^*$ | $x_2^*$ | $g(x_1^*, x_2^*)$ |
|---|---|---|---|
| 1 | 2.9924755690 | 0.4980467408 | 1.000009290398105 |
| 2 | 2.9956054090 | 0.4988096803 | 1.000003329231438 |
| 3 | 2.9980444312 | 0.4994963258 | 1.000000621200897 |
| 4 | 2.9882788059 | 0.4970790594 | 1.000022278883427 |
| 5 | 2.9687499395 | 0.4921500531 | 1.000161861260038 |
| 6 | 2.9906343815 | 0.4971046894 | 1.000021421032643 |
| 7 | 3.0066015722 | 0.5017091483 | 1.000007061393736 |
| 8 | 2.9882805940 | 0.4968260377 | 1.000023733542026 |
| 9 | 2.9995339513 | 0.4998777956 | 1.000000035792665 |
| 10 | 2.9975866675 | 0.4993637055 | 1.000000966651146 |

## b)

Below, in Table 5, the results from the batch runs with different values of $p_{\text{mut}}$ are presented.

*Table 5: Median performance for different values of $p_{\text{mut}}$.*

| $p_{\text{mut}}$ | Median performance $(F)$ |
|---|---|
| 0.00 | 0.993782439775741 |
| 0.02 | 0.999857602698811 |
| 0.05 | 0.999824574054263 |
| 0.10 | 0.999763037943086 |
| 0.20 | 0.999366037331218 |
| 0.25 | 0.999136577867517 |
| 0.35 | 0.998947754692627 |
| 0.45 | 0.999009880723805 |
| 0.60 | 0.998476643622652 |
| 0.80 | 0.998760322853456 |

The results from the above runs are also presented in a plot displayed in figure 2.

Considering the above results, we see that the optimal value of $p_{\text{mut}}$ is expected to lie around 0.02.

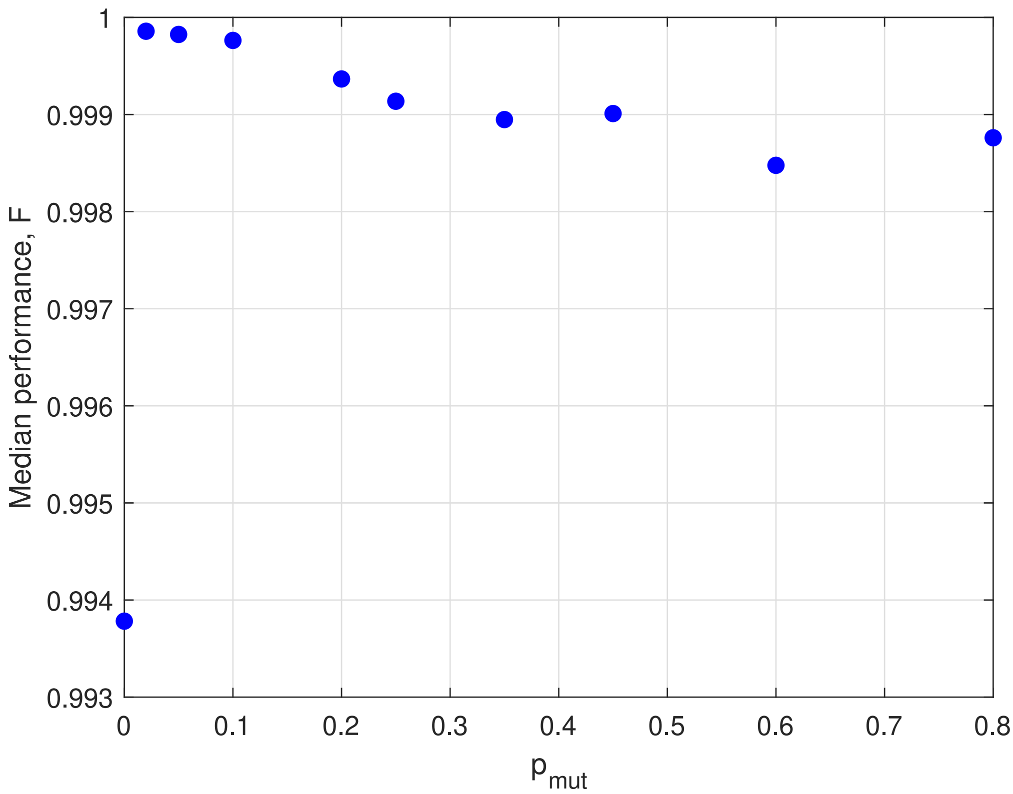


*Figure 2: Caption*

## c)

To prove that the estimated minimum $\mathbf{x}^* = (0.3, 0.5)$ from task a) is indeed a stationary point, we calculate the gradient $\nabla g(x_1, x_2)$ and insert $\mathbf{x}^*$, hoping to find that $\nabla g(\mathbf{x}^*) = \mathbf{0}$.

We have that $\nabla g(x_1, x_2) = \left[\frac{\partial g}{\partial x_1}, \frac{\partial g}{\partial x_2}\right]^T$, so we begin by finding the partial derivatives:

$$\frac{\partial g}{\partial x_1} = 2(1.5 - x_1 + x_1 x_2)(x_2 - 1) + 2(2.25 - x_1 + x_1 x_2^2)(x_2^2 - 1) + 2(2.625 - x_1 + x_1 x_2^3)(x_2^3 - 1)$$
$$\frac{\partial g}{\partial x_2} = 2(1.5 - x_1 + x_1 x_2)x_1 + 2(2.25 - x_1 + x_1 x_2^2)2x_1 x_2 + 2(2.625 - x_1 + x_1 x_2^3)3x_2^2 x_1$$

Inserting $\mathbf{x}^* = (0.3, 0.5)$, we find that $\nabla g(0.3, 0.5) = (-8.5, 3.1)^T$. This indicates that the estimate is not close to the true minimum. However, that is if we assume that the above calculations are correct.