

CHALMERS UNIVERSITY OF TECHNOLOGY

FFR105 - Stochastic Optimization Algorithms

Problem set 1

Jacopo Credi  
(910216-T396)

September 25, 2015



## Problem 1.1 - Penalty method

In this problem, we used the penalty method to find the minimum of the function

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

subject to the inequality constraint

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

i.e.  $(x_1, x_2)^T \in S$ , where  $S$  is the unit circle centered at the origin.

(1)

The penalty function  $p(\mathbf{x}; \mu)$  of the problem is

$$p(\mathbf{x}; \mu) = \mu \left( \max\{g(x_1, x_2), 0\} \right)^2 = \mu \left( \max\{x_1^2 + x_2^2 - 1, 0\} \right)^2, \quad (3)$$

thus, the problem can be transformed into an unconstrained minimization of the function

$$\begin{aligned} f_p(\mathbf{x}; \mu) &= f(x_1, x_2) + p(x_1, x_2; \mu) \\ &= (x_1 - 1)^2 + 2(x_2 - 2)^2 + \mu \left( \max\{x_1^2 + x_2^2 - 1, 0\} \right)^2. \end{aligned} \quad (4)$$

(2)

The gradient of the function  $f_p(\mathbf{x}; \mu)$  has a different form, depending on the region where it is calculated:

$$\underline{\nabla} f_p(\mathbf{x}; \mu) = \left( \frac{\partial f_p}{\partial x_1}, \frac{\partial f_p}{\partial x_2} \right)^T,$$

with

$$\begin{aligned} \frac{\partial f_p}{\partial x_1} &= \begin{cases} 2x_1 - 2, & \text{if } x_1^2 + x_2^2 - 1 \leq 0 \\ 2x_1 - 2 + 4\mu x_1 (x_1^2 + x_2^2 - 1), & \text{if } x_1^2 + x_2^2 - 1 > 0 \end{cases}, \\ \frac{\partial f_p}{\partial x_2} &= \begin{cases} 4x_2 - 8, & \text{if } x_1^2 + x_2^2 - 1 \leq 0 \\ 4x_2 - 8 + 4\mu x_2 (x_1^2 + x_2^2 - 1), & \text{if } x_1^2 + x_2^2 - 1 > 0 \end{cases}. \end{aligned}$$

(3)

The unconstrained minimum of  $f_p(\mathbf{x}; \mu = 0) = f(\mathbf{x})$  can be found by computing

$$\underline{\nabla} f = 0 \quad \Leftrightarrow \quad \begin{cases} 2x_1 - 2 = 0 \\ 4x_2 - 8 = 0 \end{cases} \quad \Leftrightarrow \quad P_0 = (1, 2)^T.$$

Point  $P_0$  is the only stationary point of the function. Let us now compute the eigenvalues of the Hessian matrix  $H$ , in order to show that the function is convex:

$$H = \begin{pmatrix} \frac{\partial^2 f_p}{\partial x_1^2} & \frac{\partial^2 f_p}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f_p}{\partial x_2 \partial x_1} & \frac{\partial^2 f_p}{\partial x_2^2} \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 4 \end{pmatrix} .$$

Since  $H$  is diagonal, the diagonal elements are its eigenvalues. They are both positive, so  $H$  is positive definite,  $f$  is (strictly) convex and since  $S$  is a convex set  $P_0 = (1, 2)^T$  is the point corresponding to the global minimum. The global minimum of  $f$  is therefore  $f(1, 2) = 0$ .

(4)

A MATLAB program was used to find the minimum of  $f_p(\mathbf{x}; \mu)$  using the method of gradient descent. The program consists of a main file `PenaltyMethod.m` which makes use of auxiliary functions `GradientDescent.m`, `Gradient.m` and `FancyPlots.m` and which can be run without requiring any input from the user, i.e. functions  $f_p$ ,  $g$ , the gradient  $\nabla f_p$  and the parameters needed by the program are hardcoded in the beginning of the main script.

The important parameters of the method are the following:

- starting point  $\mathbf{x}_0$  for the gradient descent;
- sequence of  $\mu$  values (*increasing*) for subsequent minimization phases;
- step length  $\eta$  in the gradient descent;
- threshold value  $T$  that determines when to stop the gradient descent.

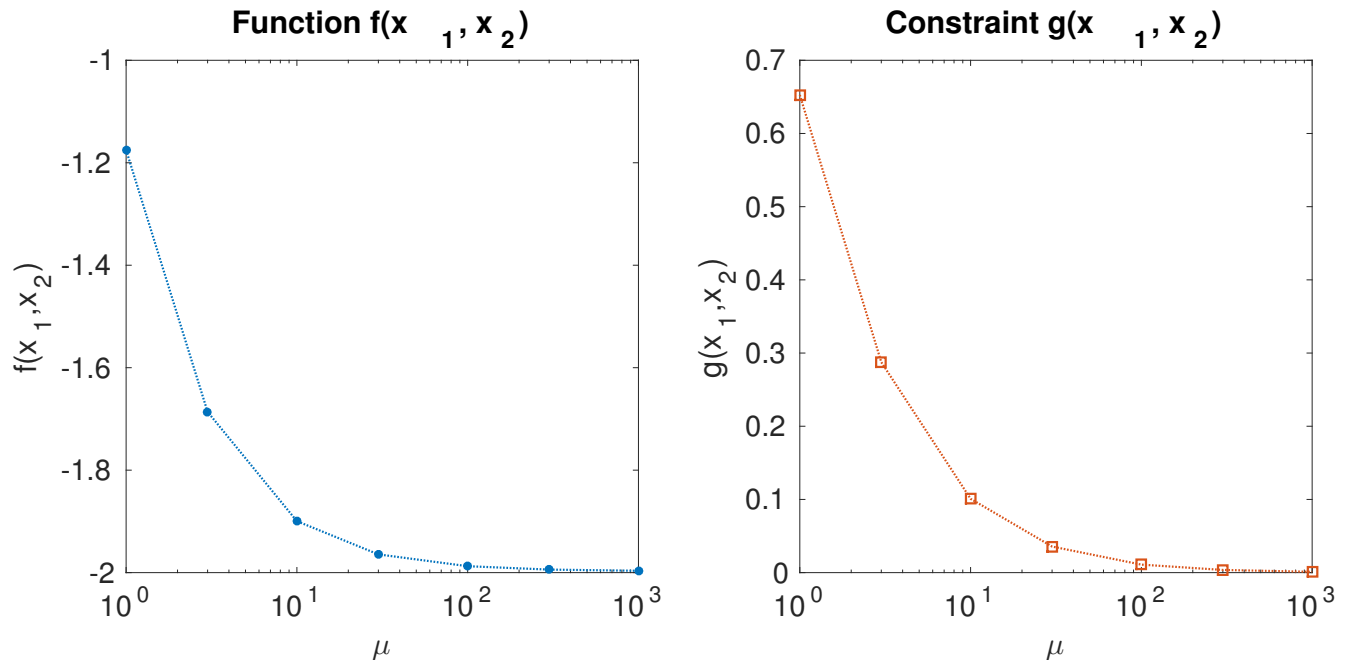
The program performs a sequence of gradient descents minimizing the function  $f_p(\mathbf{x}; \mu)$  with increasing  $\mu$  (but fixed in each minimization phase) and the output is a  $n$ -by-3 matrix, where  $n$  is the number of  $\mu$  values in the input sequence, and where each row  $(\mu^{(i)} \ x_1^{*(i)} \ x_2^{*(i)})$  contains the value taken by  $\mu$  in the  $i^{th}$  optimization phase and the corresponding position  $(x_1^{*(i)}, x_2^{*(i)})$  of the minimum of  $f_p$  found in phase  $i$ .

(5)

The implemented method was applied to minimize function  $f$  in Eq.(1), subject to the inequality constraint in Eq.(2). The unconstrained minimum  $P_0$  of  $f$  was used as starting point for the first minimization phase, i.e.  $\mathbf{x}_0 = P_0 = (1, 2)^T$ . The step length was set to  $\eta = 10^{-4}$  and the threshold to  $T = 10^{-6}$ . The used sequence of  $\mu$  values is displayed in the first column of Table 1, which is exactly the output table produced by the program. The position of the minimum clearly seems to converge to a point on the unit circle, as  $\mu$  is increased.

As shown in Fig. 1 (left panel), function  $f$  evaluations at points found by the program converge to the value -2, which can be claimed to be the solution of the constrained minimization problem. On the right panel, instead, we can see that when  $\mu \rightarrow \infty$  the constraint  $g(x_1, x_2) \leq 0$  is satisfied.

$\mu$	$x_1^*$	$x_2^*$
1	0.434	1.210
3	0.367	1.074
10	0.331	0.996
30	0.319	0.967
100	0.314	0.955
300	0.312	0.952
1000	0.312	0.951

Table 1: Output table of program `PenaltyMethod.m` with parameters specified above.Figure 1: Functions  $f$  (left panel) and  $g$  (right panel) evaluated at the points found by the subsequent optimization phases, as  $\mu$  is increased.

## Problem 1.2 - Constrained optimization

(a)

Here we are requested to find the global minimum of the function

$$f(x_1, x_2) = 4x_1^2 - x_1x_2 + 4x_2^2 - 6x_2, \quad (5)$$

on the closed set  $S$ , where  $S$  is a triangle with vertices in  $A = (0, 0)$ ,  $B = (1, 1)$ ,  $C = (0, 1)$ , using the “analytical” method.

### 1. Find stationary points

$$\nabla f = 0 \Leftrightarrow \begin{cases} \frac{\partial f}{\partial x_1} = 8x_1 - x_2 = 0 \\ \frac{\partial f}{\partial x_2} = -x_1 + 8x_2 - 6 = 0 \end{cases} \Rightarrow P_1 = \left(\frac{2}{21}, \frac{16}{21}\right) \text{ is the only stationary point.}$$

The first critical point to check,  $P_1$ , lies inside the set  $S$ .

### 2. Check the boundary $\partial S$

$$\begin{aligned} x_1 = 0 &\Rightarrow f(0, x_2) = 4x_2^2 - 6x_2 \\ \frac{\partial}{\partial x_2} f(0, x_2) &= 8x_2 - 6 \\ \frac{\partial}{\partial x_2} f(0, x_2^*) &= 0 \Leftrightarrow x_2^* = \frac{3}{4} \Rightarrow P_2 = \left(0, \frac{3}{4}\right) \text{ must also be checked.} \end{aligned}$$

$$\begin{aligned} x_2 = 1 &\Rightarrow f(x_1, 1) = 4x_1^2 - x_1 - 2 \\ \frac{\partial}{\partial x_1} f(x_1, 1) &= 8x_1 - 1 \\ \frac{\partial}{\partial x_1} f(x_1^*, 1) &= 0 \Leftrightarrow x_1^* = \frac{1}{8} \Rightarrow P_3 = \left(\frac{1}{8}, 1\right) \text{ must also be checked.} \end{aligned}$$

$$\begin{aligned} x_1 = x_2 = x &\Rightarrow f(x, x) = 7x^2 - 6x \\ \frac{\partial}{\partial x} f(x, x) &= 14x - 6 \\ \frac{\partial}{\partial x} f(x^*, x^*) &= 0 \Leftrightarrow x^* = \frac{3}{7} \Rightarrow P_4 = \left(\frac{3}{7}, \frac{3}{7}\right) \text{ must also be checked.} \end{aligned}$$

Also, corners  $P_5 = A = (0, 0)$ ,  $P_6 = B = (1, 1)$ ,  $P_7 = C = (0, 1)$  must be checked.

### 3. Evaluate found points one by one

We have that

$$\min \{f(P_1), f(P_2), f(P_3), f(P_4), f(P_5), f(P_6), f(P_7)\} = f(P_1) = -\frac{16}{7},$$

and therefore the global minimum is located in  $P_1 = \left(\frac{2}{21}, \frac{16}{21}\right)$ .

One could also have computed the eigenvalues of Hessian matrix

$$H = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{pmatrix} = \begin{pmatrix} 8 & -1 \\ -1 & 8 \end{pmatrix},$$

$$|H - \lambda I_2| = (8 - \lambda)^2 - 1 = 0 \Leftrightarrow \lambda_1 = 9, \lambda_2 = 7,$$

finding that  $H$  is positive definite, so the function  $f$  is strictly convex and therefore its minimum in the convex set  $S$  must be in the interior of  $S$  (no need to check the boundary and corners).

(b)

Here we are requested to find the minimum of the function

$$f(x_1, x_2) = 15 + 2x_1 + 3x_2 , \quad (6)$$

subject to the equality constraint

$$h(x_1, x_2) = x_1^2 + x_1x_2 + x_2^2 - 21 = 0 , \quad (7)$$

using the Lagrange multiplier method.

The Lagrange function  $\mathcal{L}(x_1, x_2, \lambda)$  is

$$\begin{aligned} \mathcal{L}(x_1, x_2, \lambda) &= f(x_1, x_2) + \lambda h(x_1, x_2) \\ &= 15 + 2x_1 + 3x_2 + \lambda (x_1^2 + x_1x_2 + x_2^2 - 21) , \end{aligned} \quad (8)$$

and we have to minimize it with respect to its arguments, by finding stationary points

$$\left\{ \begin{array}{l} \frac{\partial \mathcal{L}}{\partial x_1} = 2 + 2\lambda x_1 + \lambda x_2 = 0 \end{array} \right. \quad (9)$$

$$\left\{ \begin{array}{l} \frac{\partial \mathcal{L}}{\partial x_2} = 3 + \lambda x_1 + 2\lambda x_2 = 0 \end{array} \right. \quad (10)$$

$$\left\{ \begin{array}{l} \frac{\partial \mathcal{L}}{\partial \lambda} = x_1^2 + x_1x_2 + x_2^2 - 21 = 0 \end{array} \right. \quad (11)$$

Multiplying Eq.(9) by 2 and subtracting Eq.(10) from it yields

$$2 \times (9) - (10) = 4 + 4\lambda x_1 + 2\lambda x_2 - 3 - \lambda x_1 - 2\lambda x_2 = 1 + 3\lambda x_1 = 0 \Leftrightarrow x_1 = -\frac{1}{3\lambda} . \quad (12)$$

Similarly multiplying Eq.(10) by 2 and subtracting Eq.(9) from it yields

$$2 \times (10) - (9) = 6 + 2\lambda x_1 + 4\lambda x_2 - 2 - 2\lambda x_1 - \lambda x_2 = 4 + 3\lambda x_2 = 0 \Leftrightarrow x_2 = -\frac{4}{3\lambda} . \quad (13)$$

Comparing Eq.(12) and (13) yields  $x_2 = 4x_1$ , which can be plugged into Eq.(11) to get

$$\begin{aligned} x_1^2 + 4x_1^2 + 16x_1^2 &= 21 \\ 21x_1^2 &= 21 \Leftrightarrow x_1 = \pm 1 . \end{aligned}$$

Hence, we obtain two stationary points  $P_{1,2} = (\pm 1, \pm 4)$ . Clearly,  $f(P_1) < f(P_2)$ , thus the minimum of  $f$  subject to  $h = 0$  is  $f(P_1) = 1$ .

## Problem 1.3 - Basic GA program

(a)

A basic genetic algorithm for function optimization was implemented in MATLAB, starting from the program written during the MATLAB introduction. The program consists of a main script (`FunctionOptimization.m`) which calls other functions and which runs without any input, i.e. all parameters are hardcoded in the beginning of the script, whereas the function to minimize is hardcoded in function `EvaluateIndividual.m`.

An example of the program output on MATLAB console is displayed in Fig. 2. At the end of the GA, the program displays the maximum fitness found, the minimum value of the function and its corresponding position. Graphical output, namely the evolution of maximum fitness over the generations and a three-dimensional visualization of the population, is also produced by the program (see e.g. Fig. 3).

```
>> FunctionOptimization
Running GA...
GA completed. 5000 individuals evaluated.

Maximum fitness found: 0.333
Minimum value of g found: 3.000
Location of minimum: (-0.000 ; -1.000)
>>
```

Figure 2: Sample console output of the main script `FunctionOptimization.m`.

(b)

The implemented GA optimizer was used to find the (global) minimum value of the function

$$g(x_1, x_2) = \left(1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right) \times \left(30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\right) \quad (14)$$

in the interval  $x_1, x_2 \in [-5, 5]$ . The fitness function  $f$  was defined to be  $1/g(x_1, x_2)$ .

Each variable  $x_i$ ,  $i = 1, 2$  was encoded with 32 genes/bits ( $m = 64$  is the total chromosome length) and in the elitism step a single copy of the best individual was inserted in the new population at each generation (although the parameter  $n_c$  is modifiable in the main script).

After a few test runs, the program was run with the parameters displayed in Table 2. Compared to the parameters used in introductory session, the population size was increased to  $N = 50$  individuals, and the mutation probability was set to  $p_{\text{mut}} = 1/m$ , so that one gene per chromosome is mutated at each step, on average. The number of generations used in the introductory session ( $G = 100$ ) was estimated to be sufficient and therefore was not changed. 50 runs were performed with these parameters, storing the best solution  $(x_1^{\text{best}}, x_2^{\text{best}})$  and its fitness value at the end of each run.



Population size $N$	Number of generations $G$	Tournament selection parameter $p_{\text{tour}}$	Tournament size $j$	Crossover probability $p_c$	Mutation probability $p_{\text{mut}}$
50	100	0.75	2	0.8	0.015625

Table 2: Parameters used for GA optimization.

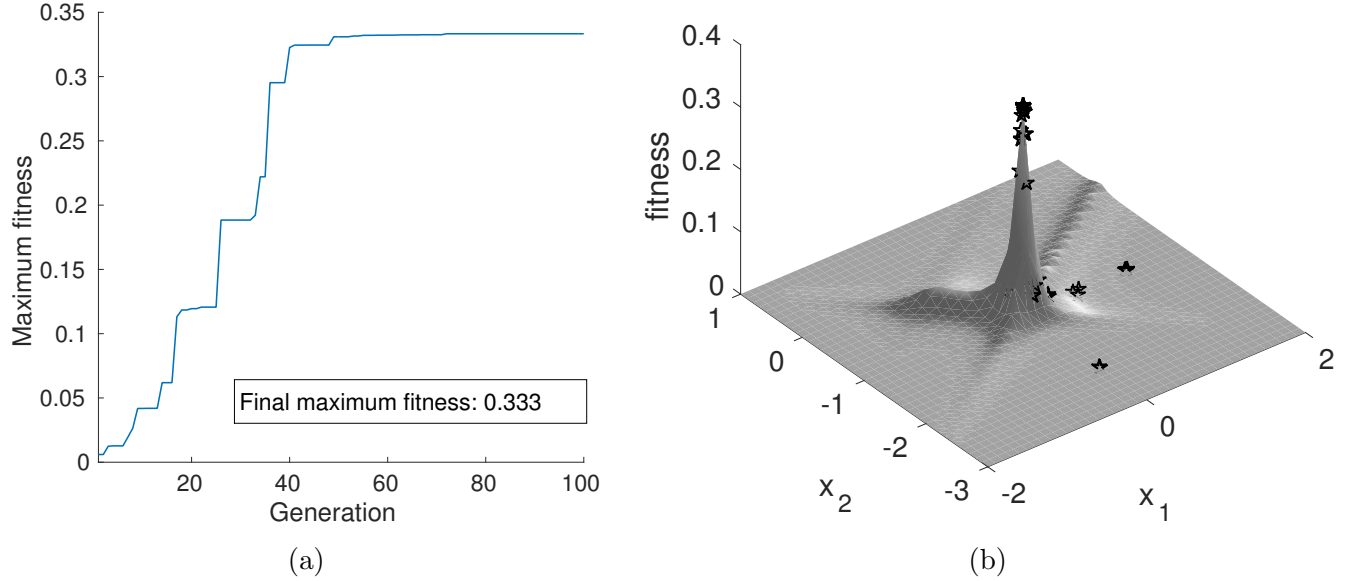


Figure 3: On the left, performance of the GA over time (number of generations). On the right, three-dimensional visualization of the final population in the fitness landscape. Data refers to the last of 50 runs performed with parameters in Table 2.

Fig. 3(a) shows the performance (maximum fitness) of the GA as function of time (or, more precisely, of the generation of the solutions population) in the last of these 50 runs. The results obtained by the GA optimizer after this first batch of runs suggest that the (global) minimum value of the function  $g$  in the set  $S = [-5, 5] \times [-5, 5]$  and its location are

$$\min_{(x_1, x_2)^T \in S} g(x_1, x_2) = 3 \quad \text{and} \quad \arg \min_{(x_1, x_2)^T \in S} g(x_1, x_2) = (0, -1) .$$

We will later see in part (c) that  $(0, -1)$  is indeed a stationary point of the function  $g$ . Table 3, instead, displays some basic statistics relative to the set of maximum fitness values  $F_{\text{max}}$  obtained in each run. We will now investigate briefly how changing some of the parameters in Table 2 affects these statistics.

Sample mean $\bar{F}_{\text{max}}$	Sample standard deviation $s(F_{\text{max}})$	Sample minimum value $\min(F_{\text{max}})$	Sample maximum value $\max(F_{\text{max}})$
0.295	0.090	0.012	0.333

Table 3: Basic statistics of a sample of 50 values of the maximum fitness obtained at the end of each of 50 runs with parameters in Table 2.

### Effect of selection methods

The effect of different parameters (tournament selection parameter  $p_{\text{tour}}$  and tournament size  $j$ ) affecting the selection phase was investigated by carrying out 50 runs with different values of  $p_{\text{tour}}$  and  $j$  and leaving other parameters unchanged (as in Table 2). Maximum fitness statistics in Table 4 show that although (at least) one optimal solution is found for each parameter set (see maximum values in the last column), the GA performance decreases as the tournament size is increased from 2 to 5. With such a small population, elitism is probably sufficient to ensure that good solutions survive the selection phase, and large tournaments impair the exploration of new solutions. As for the tournament selection parameter  $p_{\text{tour}}$ , its effect is less clear, as the standard deviation of the samples is too high to draw conclusions about it.

Selection parameters		$\bar{F}_{\max}$	$s(F_{\max})$	$\min(F_{\max})$	$\max(F_{\max})$
$j = 2$	$p_{\text{tour}} = 0.7$	0.276	0.113	0.012	0.333
$j = 2$	$p_{\text{tour}} = 0.8$	0.283	0.099	0.011	0.333
$j = 2$	$p_{\text{tour}} = 0.9$	0.253	0.119	0.011	0.333
$j = 5$	$p_{\text{tour}} = 0.7$	0.230	0.128	0.005	0.333
$j = 5$	$p_{\text{tour}} = 0.8$	0.184	0.156	0.003	0.333
$j = 5$	$p_{\text{tour}} = 0.9$	0.210	0.138	0.008	0.333

Table 4: Maximum fitness basic statistics for different parameters governing the selection phase of the GA. For other parameters see Table 2.

### Effect of crossover probability

The effect of different crossover probabilities  $p_c$  was also investigated by carrying out 50 runs with different values of  $p_c$  and leaving other parameters unchanged (see Table 2). Maximum fitness statistics in Table 5 show that crossover is an important phase allowing the GA to evolve better solutions. In fact, the mean value of the maximum fitness increases monotonically as  $p_c$  increases, except for the last case, in which  $p_c = 1$ . This might suggest that allowing selected individuals to replicate themselves unchanged with a small probability can avoid over-exploitation.

Crossover probability	$\bar{F}_{\max}$	$s(F_{\max})$	$\min(F_{\max})$	$\max(F_{\max})$
$p_c = 0.00$	0.286	0.084	0.012	0.333
$p_c = 0.25$	0.299	0.088	0.011	0.333
$p_c = 0.50$	0.304	0.089	0.011	0.333
$p_c = 0.80$	0.304	0.079	0.011	0.333
$p_c = 0.90$	0.313	0.077	0.012	0.333
$p_c = 1.00$	0.286	0.107	0.012	0.333

Table 5: Maximum fitness basic statistics for different crossover probabilities. For other parameters see Table 2.

### Effect of mutation probability

The effect of different mutation probabilities  $p_{\text{mut}}$  was also investigated by carrying out 50 runs with different values of  $p_{\text{mut}}$  and leaving other parameters unchanged (see Table 2). Maximum fitness statistics in Table 6 show that if mutations are completely inhibited the GA performance dramatically drops, as the population is very likely to get stuck in a local optimum. If  $p_{\text{mut}}$  is increased, GA performance gradually increases and reaches its best value for  $p_{\text{mut}} = 3/m$ , probably corresponding to a good compromise between exploration and exploitation. Above this value, the excessive randomness seems to negatively affect the GA performance as exploitation is impaired.

Mutation probability	$\bar{F}_{\text{max}}$	$s(F_{\text{max}})$	$\min(F_{\text{max}})$	$\max(F_{\text{max}})$
$p_{\text{mut}} = 0$	0.089	0.095	0.001	0.332
$p_{\text{mut}} = 1/2m$	0.251	0.112	0.011	0.333
$p_{\text{mut}} = 1/m$	0.289	0.104	0.011	0.333
$p_{\text{mut}} = 3/m$	0.319	0.063	0.012	0.333
$p_{\text{mut}} = 10/m$	0.280	0.055	0.126	0.333
$p_{\text{mut}} = 1$	0.188	0.122	0.010	0.333

Table 6: Maximum fitness basic statistics for different mutation probabilities.

(c)

The point  $(x_1^*, x_2^*)^T = (0, -1)^T$  found by the GA can be proven to be a stationary point of the function  $g$ . Let us first rewrite  $g$  as

$$\begin{aligned}
 g(x_1, x_2) &= \left(1 + \underbrace{(x_1 + x_2 + 1)^2}_{g_1(x_1, x_2)} \underbrace{(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)}_{g_2(x_1, x_2)}\right) \times \\
 &\quad \left(30 + \underbrace{(2x_1 - 3x_2)^2}_{g_3(x_1, x_2)} \underbrace{(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)}_{g_4(x_1, x_2)}\right) \\
 &= \left(1 + g_1(x_1, x_2)g_2(x_1, x_2)\right) \left(30 + g_3(x_1, x_2)g_4(x_1, x_2)\right), \tag{15}
 \end{aligned}$$

and note that

$$g_1(x_1^*, x_2^*) = 0, \quad g_2(x_1^*, x_2^*) = 36, \quad g_3(x_1^*, x_2^*) = 9, \quad g_4(x_1^*, x_2^*) = -3. \tag{16}$$

Now, by taking the derivative of Eq.(15) with respect to  $x_i$ ,  $i = 1, 2$ , we have<sup>1</sup>

$$\begin{aligned}
 \frac{\partial g}{\partial x_i} &= (30 + g_3 g_4) \frac{\partial}{\partial x_i}(1 + g_1 g_2) + (1 + g_1 g_2) \frac{\partial}{\partial x_i}(30 + g_3 g_4) \\
 &= (30 + g_3 g_4) \left(g_2 \frac{\partial g_1}{\partial x_i} + g_1 \frac{\partial g_2}{\partial x_i}\right) + (1 + g_1 g_2) \left(g_4 \frac{\partial g_3}{\partial x_i} + g_3 \frac{\partial g_4}{\partial x_i}\right) \tag{17}
 \end{aligned}$$

Now, evaluating this generic derivative at point  $(x_1^*, x_2^*)^T = (0, -1)^T$  and using values displayed in (16), Eq.(17) simplifies to

$$\left. \frac{\partial g}{\partial x_i} \right|_{(0, -1)} = 108 \left. \frac{\partial g_1}{\partial x_i} \right|_{(0, -1)} - 3 \left. \frac{\partial g_3}{\partial x_i} \right|_{(0, -1)} + 9 \left. \frac{\partial g_4}{\partial x_i} \right|_{(0, -1)} \tag{18}$$

<sup>1</sup>From now on the cumbersome notation  $g_j(x_1, x_2)$  is dropped and substituted by  $g_j$ .

We can now easily compute the derivatives with respect to  $x_1$  and  $x_2$  and evaluated at  $(x_1^*, x_2^*)^T$ :

$$\begin{aligned}\frac{\partial g}{\partial x_1} \Big|_{(0,-1)} &= 216 (x_1 + x_2 + 1) \Big|_{(0,-1)} - 12 (2x_1 - 3x_2) \Big|_{(0,-1)} + 9(-32 + 24x_1 - 36x_2) \Big|_{(0,-1)} \\ &= 0 - 36 + 36 \\ &= 0 ,\end{aligned}$$

$$\begin{aligned}\frac{\partial g}{\partial x_2} \Big|_{(0,-1)} &= 216 (x_1 + x_2 + 1) \Big|_{(0,-1)} + 18 (2x_1 - 3x_2) \Big|_{(0,-1)} + 9 (48 - 36x_1 + 54x_2) \Big|_{(0,-1)} \\ &= 0 + 54 - 54 \\ &= 0 .\end{aligned}$$

We have  $\underline{\nabla} g(0, -1) = (0, 0)^T$ , hence the found point is stationary.