Homework Problems 1 - Report

Name: Robert Grzelka, Personnumber: 911009C759

September 2017

Contents

1. Problem 1.1 - Penalty Method

1.1 Problem Description

In this problem, goal was to nd the minimum of the objective, constrained, binomial function

$$f(x_1, x_2) = (x_1 - 1)^2 + 2 * (x_2 - 2)^2 \tag{1.1}$$

that is subject to the inequality constraint function

$$g(x_1, x_2) = x_1^2 + x_2^2 - 1 \leq 0. \tag{1.2}$$

Problem may be presented graphically by plot in 3 dimensions as below:

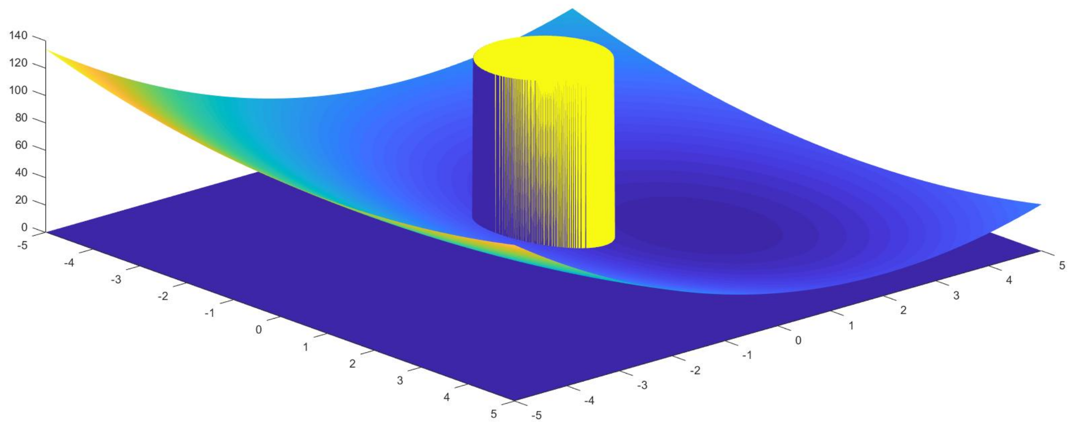


Figure 1.1: Objective Function with constraints.

and with vertical projection on the plane (x_1, x_2)
Following "Home Problems 1 Instruction" this task include steps that are listed below as copy:

1. Define (and specify clearly, in your report, as a function of x_1, x_2, and μ) the function $f_p(x; \mu)$, consisting of the sum of $f(x_1, x_2)$ and the penalty term $p(x; \mu)$.

2. Compute (analytically) the gradient $\nabla f_p(x; \mu)$, and include it in your report.

3. Find and report the unconstrained minimum (i.e. for $\mu = 0$) of the function. This point will be used as the starting point for gradient descent.

4. Write a Matlab program for solving the unconstrained problem of nding the minimum of $f_p(x; \mu)$ using the method of gradient descent.

   (a) A main file PenaltyMethod.m (that calls the other functions, generates and prints output etc. etc.). This program should not require any input, i.e. to run it, one should only need to write PenaltyMethod in Matlab, without having to specify any input parameters. The necessary parameters may be hardcoded in PenaltyMethod.m; see also below.

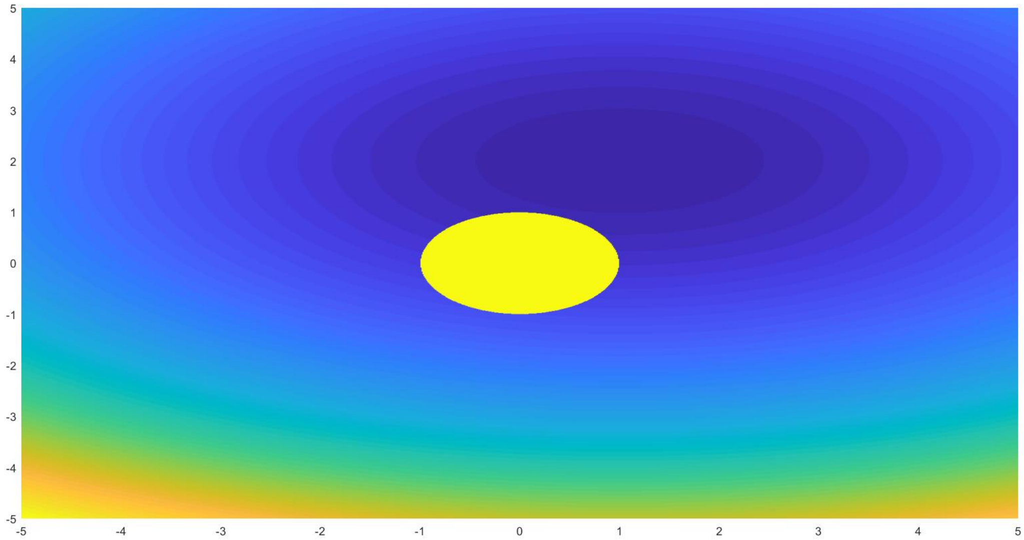


Figure 1.2: Objective Function with constraints with vertical view.

(b) A function (in a separate file, GradientDescent.m), which takes the starting point $\vec{x}_{(0)}$ (as a vector with two elements), the value of μ, the step length (for gradient descent) η, and a threshold T (see below) as input, and carries out gradient descent until the modulus of the gradient, $|\nabla f_p(\vec{x};\mu)|$, drops below the threshold T. Use the unconstrained minimum at the starting point; see above.

(c) A function Gradient (in a separate le, Gradient.m) which takes as input the values of x_1, x_2, and μ, and returns the gradient of $f_p(\vec{x};\mu)$ (a vector with two elements). Note: You may hardcode the gradient in this method, i.e. you do not need to write a general method for nding the gradient. However, your method should make use of the analytical gradient, computed in Step 2 above. You should not use a numerical approximation of the gradient.

5. Run the program for a suitable sequence of μ values (which you may hard-code in PenaltyMethod.m). Select a suitable (small) value for the step length η, and specify it clearly, along with the sequence of μ values, in your report. Example of suitable parameter values: $\eta = 0.0001$, $T = 10e - 1$, sequence of μ values: 1, 10, 100, 1000.

Program in Matlab is not required to be reported, so only instruction of "how to" run it will be attached and necessary results with graphic output. Next sections will be describing steps used for solving this task.

1.2 Step 1

Using the penalty method (pp. 30-33 in the course book) we rewrite problem as unconstrained objective function

$$f_p(x_1, x_2, \mu) = f(x_1, x_2) + p(x_1, x_2, \mu) \tag{1.3}$$

where $f(x_1, x_2)$ is constrained objective function and

$$p(x_1, x_2, \mu) = \mu * max\{0, x_1^2 + x_2^2 - 1\}^2 \tag{1.4}$$

is penalty term (function), what could be rewritten as

$$p(x_1, x_2, \mu) = \mu * \begin{cases} a(x_1, x_2)^2 & \text{if} \quad a(x_1, x_2) >= 0 \\ 0 & \text{otherwise} \end{cases} \tag{1.5}$$

and where

$$a(x_1, x_2) = x_1^2 + x_2^2 - 1. \tag{1.6}$$

1.3 Step 2

Solution with penalty method algorithm included gradient descent, where gradient is of unconstrained objective function and it has form

$$\nabla f_p(x_1, x_2, \mu) = \nabla f(x_1, x_2) + \nabla p(x_1, x_2, \mu) \tag{1.7}$$

Gradient of objective function

$$\nabla f(x_1, x_2) = \left(\frac{\partial f(x_1, x_2)}{\partial x_1}, \frac{\partial f(x_1, x_2)}{\partial x_2} \right) \tag{1.8}$$

contains partial derivatives over variables x_1, derived as

$$\frac{\partial f(x_1, x_2)}{\partial x_1} = 2 * (x_1 - 1) \tag{1.9}$$

and over x_2, derived as

$$\frac{\partial f(x_1, x_2)}{\partial x_2} = 4 * (x_2 - 2). \tag{1.10}$$

Also, gradient of penalty function is of a form

$$\nabla p(x_1, x_2, \mu) = \mu * \left(\frac{\partial p(x_1, x_2, \mu)}{\partial x_1}, \frac{\partial p(x_1, x_2, \mu)}{\partial x_2} \right) \tag{1.11}$$

for which partial derivatives over variables x_1 and x_2 are derived (μ is treated as constant parameter, not to be derived for) respectively as

$$\frac{\partial p(x_1, x_2)}{\partial x_1} = 4 * x_1 * (x_1^2 + x_2^2 - 1) \tag{1.12}$$

and

$$\frac{\partial p(x_1, x_2)}{\partial x_1} = 4 * x_2 * (x_1^2 + x_2^2 - 1). \tag{1.13}$$

1.4 Step 3

When $\mu = 0$ unconstrained objective function $f_p(x_1, x_2; \mu)$ is reduced to objective function $f(x_1, x_2)$. For which unconstrained minimum could be find in critical, stationary point that by definition happens where $\nabla f(x_1, x_2) = 0$.

From partial derivatives system obtained in "Step 2"

$$\begin{cases} \frac{\partial f(x_1,x_2)}{\partial x_1} = 2 * (x_1 - 1) = 0 \\ \frac{\partial f(x_1,x_2)}{\partial x_2} = 4 * (x_2 - 2) = 0 \end{cases}$$

we get $x_1 = 1$ and $x_2 = 2$ what is unconstrained minimum of $f_p(x_1, x_2)$ taken as starting point $\vec{x_{(0)}} = (1, 2)$ for gradient descent.

1.5 Instruction of usage program PenaltyMethod

Program for given problem is already set with hardcoded input parameters. After end of computation program gives output and presents solution graphically in 3d figure.

Instruction of running program:

1. open Matlab
2. run PenaltyMethod.m with input given as already is in file
3. wait for program to finish computation
4. check output from command window and look at graphic output

Describing briefly, program (after setting constant parameters) iterate for loop as follows:

1. choose next number from penalty parameters μ sequence and in first iteration start from unconstrained minima stationary point $\vec{x_{(j=0)}}$;
2. calculate gradient for of $f_p(\vec{x};\mu)$ for above values;
3. if modulus of gradient vector for function $f_p(\vec{x};\mu)$ with variables $\vec{x_{(j=0)}}$ is:
   (a) lesser than set threshold then go to step (4.) and set new starting point as current $x_{(j)}$
   (b) otherwise update starting point point by term $\vec{x_{(j)}} -> \vec{x_{(j)}} - \eta * \nabla f_p(\vec{x_{(j)}};\mu)$ following gradient direction with step length η and go to step (2.);
4. repeat steps (1.), (2.), (3.) until μ will give Inf results from gradient, then previous value of μ will give result with smallest error;

1.6 Hardcoded Input Parameters

Following instruction, input parameters included in PenaltyMethod file are as below

- step length - $\eta = 0.0001$
- threshold - $T = 10e - 1$
- penalty parameter sequence - $\mu = [1, 10, 100, 1000]$.

1.7 Results

Output results of minima points x_1, x_2 for sequence of penalty parameters μ are presented in table 1.1. Also it is possible to present corresponding $f_p(\vec{x};\mu)$ values for each solution like in table 1.1. Finally graphical representation of penalty method solution is on figure 2.2.6.

μ	x_1	x_2
0	1.000	2.000
1	0.434	1.210
10	0.331	0.996
100	0.314	0.955
1000	0.312	0.951
1249	0.312	0.951
1250	NaN	NaN

Table 1.1: OUTPUT 1: Minima points for μ sequence

μ	x_1	x_2	$f_p(\vec{x};\mu)$
0	1.000	2.000	0.000
1	0.434	1.210	1.994
10	0.331	0.996	2.567
100	0.314	0.955	2.666
1000	0.312	0.951	2.677
1249	0.312	0.951	2.677
1250	NaN	NaN	NaN

Table 1.2: OUTPUT 2: Minima points and function $f_p(\vec{x};\mu)$ values for μ sequence

1.8 Conclusions

During run program may inform about error:
"...error in GradientDescent.m (line 27) : $\|\nabla * F_p(x_1, x_2, \mu = 1250)\| = NaN$".

During this case, error pop out for too big value $\mu = 1250$ of penalty parameter, from where was found that biggest penalty parameter giving successful run is $\mu = 1249$. Also for $\mu = 1249$ solution gives best result reaching boundary of valid domain with smallest deviation compared to smaller values of penalty parameter. With smaller values of penalty parameter, points received from gradient descent do not reach valid domain boundary (or reach with deviation). Bigger value of penalty parameter gives smaller deviation, so one may conclude that it is always desired to find value of penalty parameter as big as possible.

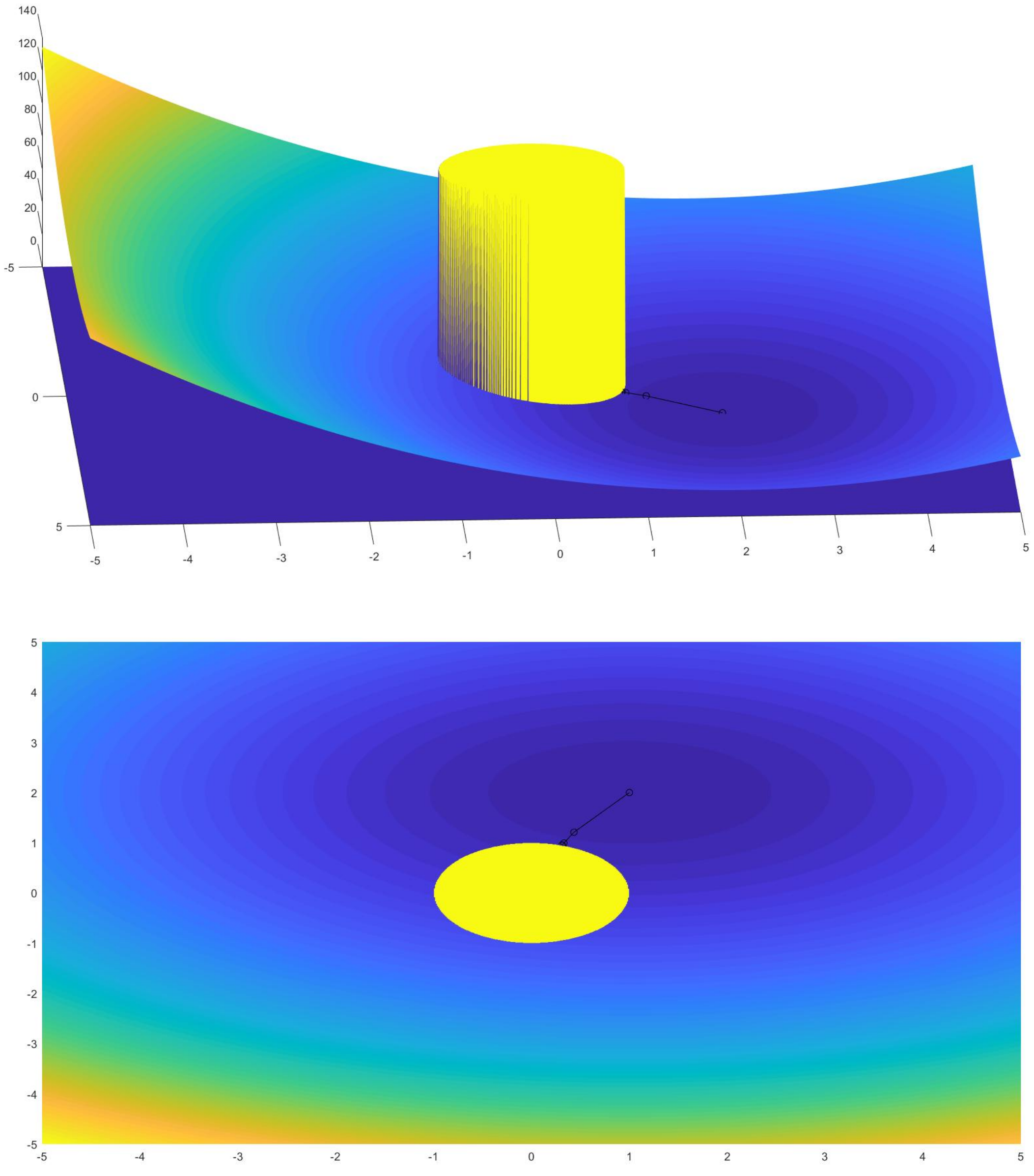


Figure 1.3: Objective Function with constraints and penalty method minima points (bottom picture with vertical view on plane (x_1, x_2))

2. Problem 1.2 - Constrained Optimization

2.1 Problem Description

This task is split on two parts a) and b). Goal of part a) is to find global minimum of objective function constrained by closed set using analytical method, and goal of part b) is to determine minimum and corresponding function value of constrained objective function using Lagrange Multipler Method.

2.2 Part a) - Analytical Method

2.2.1 Problem Introduction

As described above problem is to find a global minimum of objective function:

$$f(x_1, x_2) = 4 * x_1^2 - x_1 * x_2 + 4 * x_2^2 - 6 * x_2 \tag{2.1}$$

subject to closed set S defined as triangle by corner points located at (0,0), (0,1) and (1,1). Describing problem graphically, one may present it as on figure 2.2.1.

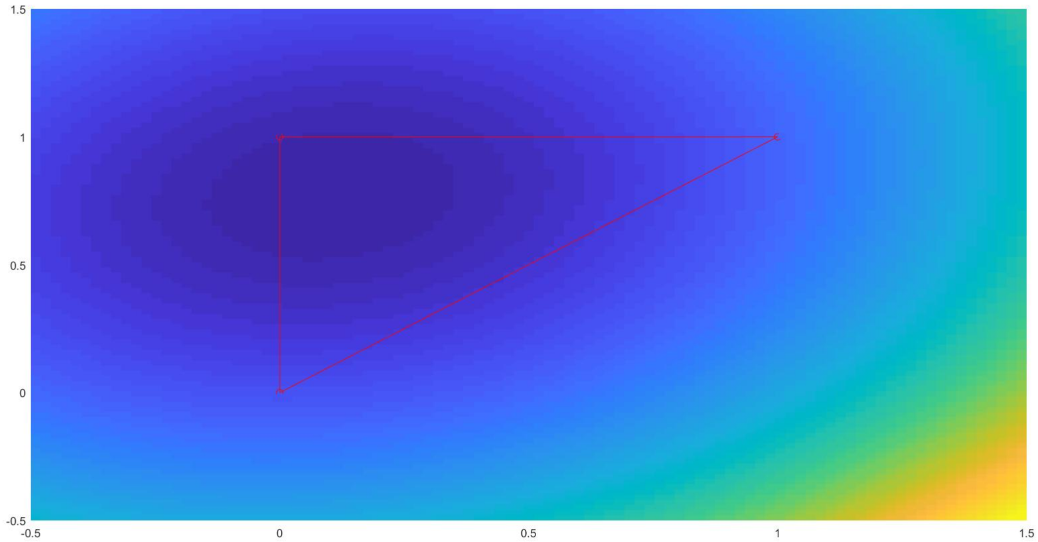


Figure 2.1: Objective Function $f(x_1, x_2) = 4 * x_1^2 - x_1 * x_2 + 4 * x_2^2 - 6 * x_2$ and closed set S

2.2.2 Solution Methodology

Steps to solve this problem are as follows:

1. First find stationary points $\vec{x}^*_{(SP)}$ of objective function on interior of closed set S. This is done by gradient of objective function compared to zero.

2. Then find stationary points $\vec{x}^*_{(SP)}$ on boundary ∂S of closed set S. For this use objective function to obtain functions of lines between each pair of corner points of triangle describing closed set S. Then get gradients for set of this line functions and solve for gradients equal to zero.

3. Next find stationary points $\vec{x}^*_{(SP)}$ on corner points of triangle of closed set S.

4. Lastly for each pair of stationary points $\vec{x}^*_{(SP)}$ select point which will give minimum objective function value. This is global minimum and solution to problem.

2.2.3 Stationary Points $\vec{x}^*_{(SP)}$ of $f(\vec{x})$ on closed set S

To find stationary point of function, by definition we compare gradient of this function to zero

$$\nabla f(x_1, x_2) = \left(\frac{\partial f(x_1, x_2)}{\partial x_1}, \frac{\partial f(x_1, x_2)}{\partial x_2} \right) = 0 \tag{2.2}$$

$$\begin{cases} \frac{\partial f(x_1,x_2)}{\partial x_1} = 8 * x_1 - x_2 + 0 - 0 = 0 \\ \frac{\partial f(x_1,x_2)}{\partial x_2} = 0 - x_1 + 8 * x_2 - 6 = 0 \end{cases} \tag{2.3}$$

then solving this simple equation system

$$\begin{cases} 8 * x_1 - x_2 + 0 - 0 = 0 & \Leftrightarrow & x_2 = 8 * x_1 & \Leftrightarrow & x_2 = 48/63 \\ x_1 + 8 * (8 * x_1) - 6 = 0 & \Leftrightarrow & x_1 = 6/63 \end{cases} \tag{2.4}$$

one finds stationary point $\vec{x}^*_{(SP1)} = (6/63, 48/63)$ in interior of closed set domain over objective function.

2.2.4 Stationary Points $\vec{x}^*_{(SP)}$ of $f(\vec{x})$ on closed set boundary ∂S

To obtain stationary points on boundary of closed set it is sufficient to define 3 lines between 3 pairs of corner points (0,0), (0,1) and (1,1) creating triangle edges what is geometrical representation of closed set S boundary.

Line from (0,0) to (0,1)

Line connecting points (0,0) and (0,1) is deduced as follows:

$$\begin{cases} x_1 = 0; \\ 0 < x_2 < 1; \end{cases} \tag{2.5}$$

Binomial objective function $f(x_1, x_2)$ for this values take form of monomial function $f(x_1 = 0, 0 < x_2 < 1) = f(0 < x_2 < 1)$:

$$f(x_2) = 4 * 0^2 - 0 * x_2 + 4 * x_2^2 - 6 * x_2 \quad \forall x_2 \in [0, 1] \tag{2.6}$$

for which gradient is reduced to first derivative and compared to zero

$$f'(x_2) = 8 * x_2 - 6 = 0 \tag{2.7}$$

and solved, gives stationary point $\vec{x}^*_{(SP2)} = (0, 6/8)$.

Line from (0,1) to (1,1)

Line between pair of corners (0,1) and (1,1) is similarly:

$$\begin{cases} 0 < x_1 < 1; \\ x_2 = 1; \end{cases} \tag{2.8}$$

Binomial objective function $f(x_1, x_2)$ for this values take form of monomial function $f(0 < x_1 < 1, x_2 = 1) = f(0 < x_1 < 1)$:

$$\begin{aligned} f(x_1) &= 4 * x_1^2 - x_1 * 1 + 4 * 1^2 - 6 * 1 \\ &= 4 * x_1^2 - x_1 + -2 \end{aligned} \tag{2.9}$$

where $f(x_1) \quad \forall x_1 \in [0, 1]$ and for which gradient is reduced to first derivative, then compared to zero

$$f'(x_2) = 8 * x_1 - 1 = 0 \tag{2.10}$$

Solving this equality, gives stationary point $\vec{x}^*_{(SP3)} = (1/8, 1)$.

Line from (0,0) to (1,1)

Lastly line from point (0,0) to point (1,1) is defined:

$$\begin{cases} 0 < x_1 < 1; \\ 0 < x_2 < 1; \end{cases} \tag{2.11}$$

or simplifying

$$\begin{cases} x_1 = x_2; \\ x_2; \end{cases} \tag{2.12}$$

Binomial objective function $f(x_1, x_2)$ for this values take form of monomial function $f(x_2)$:

$$\begin{aligned} f(x_2) &= 4 * x_2^2 - x_2 * x_2 + 4 * x_2^2 - 6 * x_2 \\ &= 7 * x_2^2 - 6 * x_2 \end{aligned} \tag{2.13}$$

where $f(x_2) \quad \forall x_2 \in [0, 1]$ and for which gradient is reduced to first derivative, then compared to zero

$$f'(x_2) = 14 * x_2 - 6 = 0 \tag{2.14}$$

Solving this equality, gives stationary point $\vec{x}^*_{(SP4)} = (3/7, 3/7)$.

2.2.5 Stationary Points $\vec{x}^*_{(SP)}$ of $f(\vec{x})$ on closed set S corner points (0,0), (0,1) and (1,1)

Stationary Point on corner (0,0)

$$f(x_1 = 0, x_2 = 0) = 0 \tag{2.15}$$

so gradient $f'(x_1 = 0, x_2 = 0) = 0$ and $\vec{x}^*_{(SP5)} = (0, 0)$

Stationary Point on corner (0,1)

$$f(x_1 = 0, x_2 = 1) = 4 * 0 - 0 * 1 + 4 * 1 - 6 * 1 = -2 \tag{2.16}$$

so gradient $f'(x_1 = 0, x_2 = 1) = 0$ and $\vec{x}^*_{(SP6)} = (0, 1)$

Stationary Point on corner (1,1)

$$f(x_1 = 1, x_2 = 1) = 4 * 1 - 1 * 1 + 4 * 1 - 6 * 1 = 1 \tag{2.17}$$

so gradient $f'(x_1 = 1, x_2 = 1) = 0$ and $\vec{x}^*_{(SP7)} = (1, 1)$

2.2.6 Results

Grouping acquired stationary points and corresponding to them function values to table 2.1, it is easy to distinguish that global minima is at point (2/21,16/21) with value -2.285.

n	x_1	x_2	$f(\vec{x})$
1	2/21	16/21	-2.285
2	0	6/8	-2.25
3	1/8	1	-2.0625
4	3/7	3/7	-1.285
5	0	0	0
6	0	1	-2
7	1	1	1

Table 2.1: Stationary Points and corresponding objective function values obtained by Analytical Method

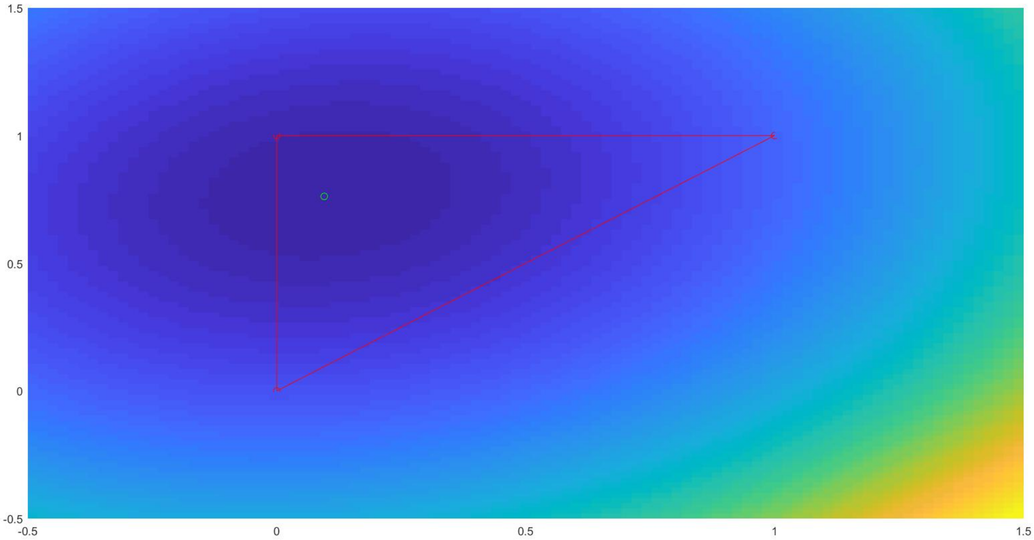


Figure 2.2: Stationary Point (marked with circle) of global minima of objective function in closed set S)

2.2.7 Conclusions

Analytical Method is based on properties of gradient of continuous differentable function. Its methodology is simple, depends on obtaining stationary points and value of this points in objective function. It may be seen that obtained global minima of function constrained by set S is equal to unconstrained function flobal minimum. From this one may deduce, that it is sufficient to find global minima of function and if it is in valid domain of closed set, this minima would give solution to problem. If its not, one need to search for other stationary points, line in corners of boundary or on boundary.

2.3 Part b) - Lagrange Multipler Method

2.3.1 Problem Introduction

In this task one has to use the Lagrange multiplier method (pp. 25-28 in the course book) to determine the minimum $(x_1^*, x_2^*)^T$ (as well as the corresponding function value) of the function $f(x_1, x_2) = 15 + 2 * x_1 + 3 * x_2$ subject to the equality constraint $h(x_1, x_2) = x_1^2 + x_1 * x_2 + x_2^2 - 21 = 0$. Problem is represented graphically on figures 2.3.1.

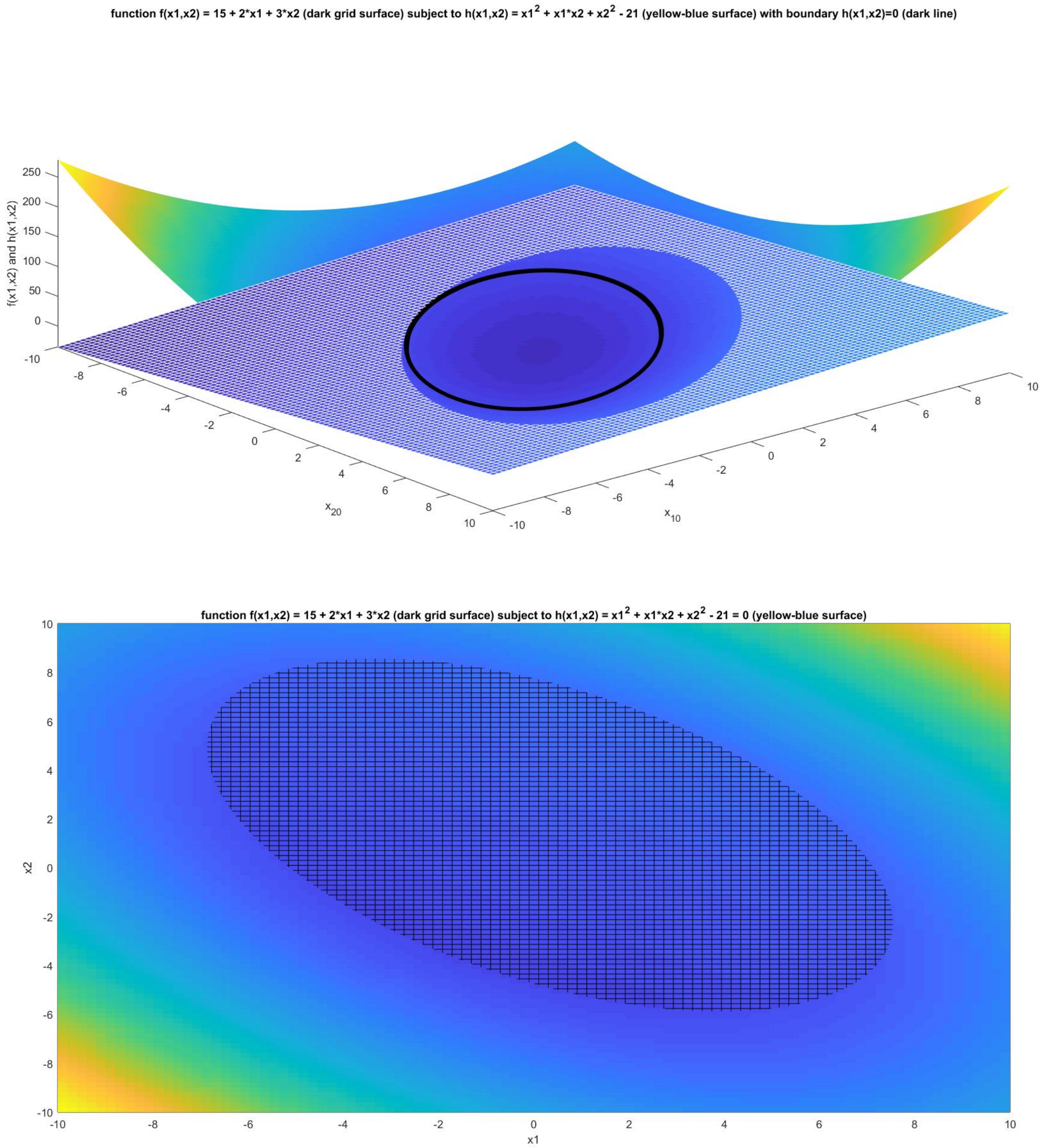


Figure 2.3: Objective function $f(x_1, x_2) = 15 + 2 * x_1 + 3 * x_2$ with equality constraint $h(x_1, x_2) = x_1^2 + x_1 * x_2 + x_2^2 - 21 = 0$ (bottom picture in vertical view).

This task is solved as follows.

1. First form problem as Lagrange Multipler unconstrained function, transforming objective function and constraints.

2. Next compare gradient of unconstrained function obtained in previous step.

3. Thirdly solve for variables x_1, x_2 and if these variables happens to be in denominator go to step (4.) otherwise go to step (5.)

4. Compare denominator to zero and solve for variables. Check if in this point function have optima.
5. Select point with maximum/minimum function value as our maximum/minimum in question of problem.

2.3.2 Lagrange Multipler unconstrained function

Rewriting objective function and its constraint to Lagrange form

$$L(\vec{x}, \lambda) = f(\vec{x}) + \lambda * h(\vec{x}) \tag{2.18}$$

one obtains:

$$\begin{aligned} L(x_1, x_2, \lambda) &= f(x_1, x2) + \lambda * h(x_1, x_2) \\ &= 15 + 2x_1 + 3x_2 + \lambda * (x_1^2 + x_1 x_2 + x_2^2 - 21) \end{aligned} \tag{2.19}$$

what is graphically as on figures (for different values of λ)

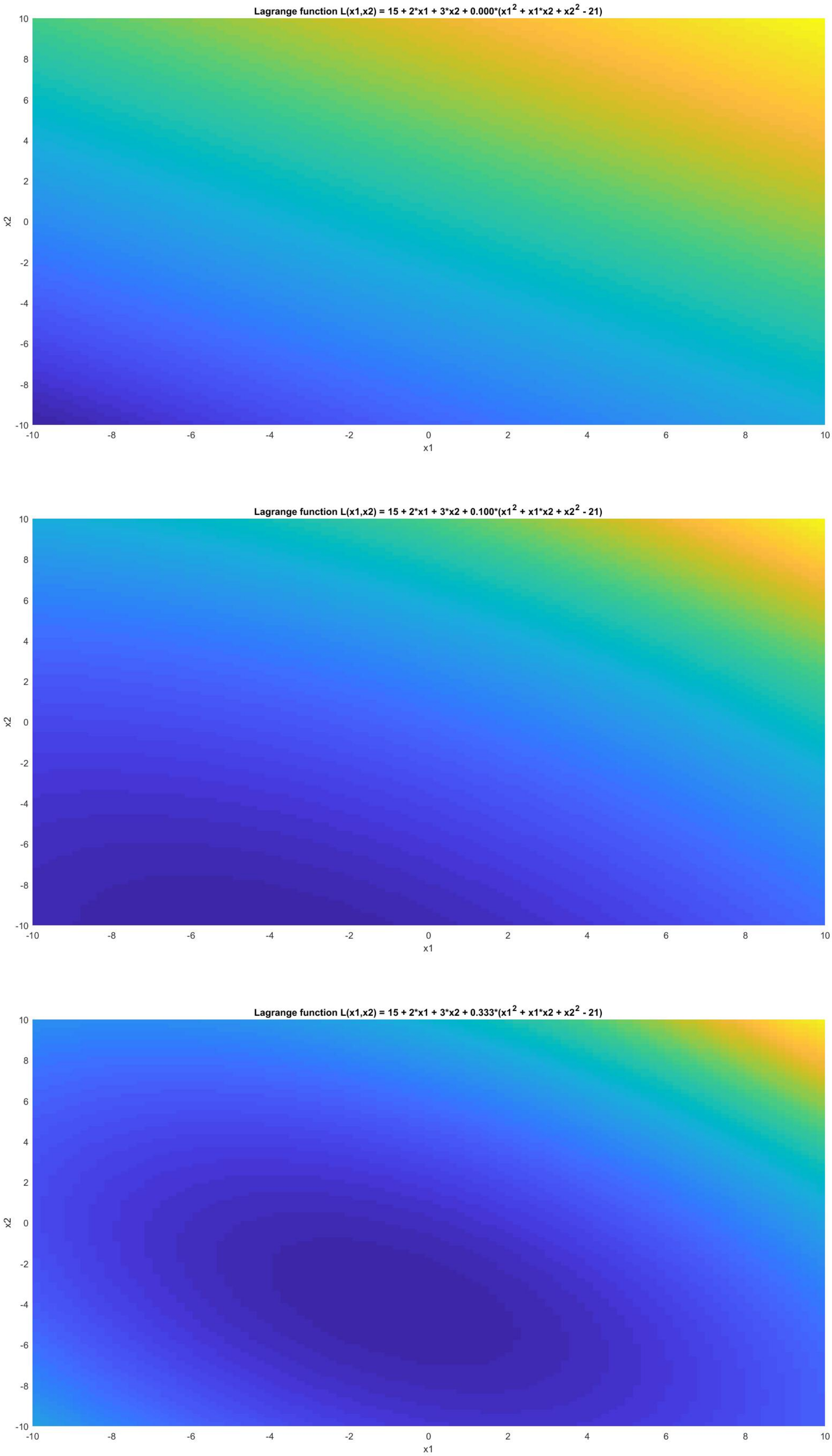


Figure 2.4: Lagrange multiplier function $L(x_1, x_2, \lambda) = f(x_1, x2) + \lambda * h(x_1, x_2)$ for sequence of λ values $[0, 1/10, 1/3]$ with vertical view on plane (x_1, x_2).

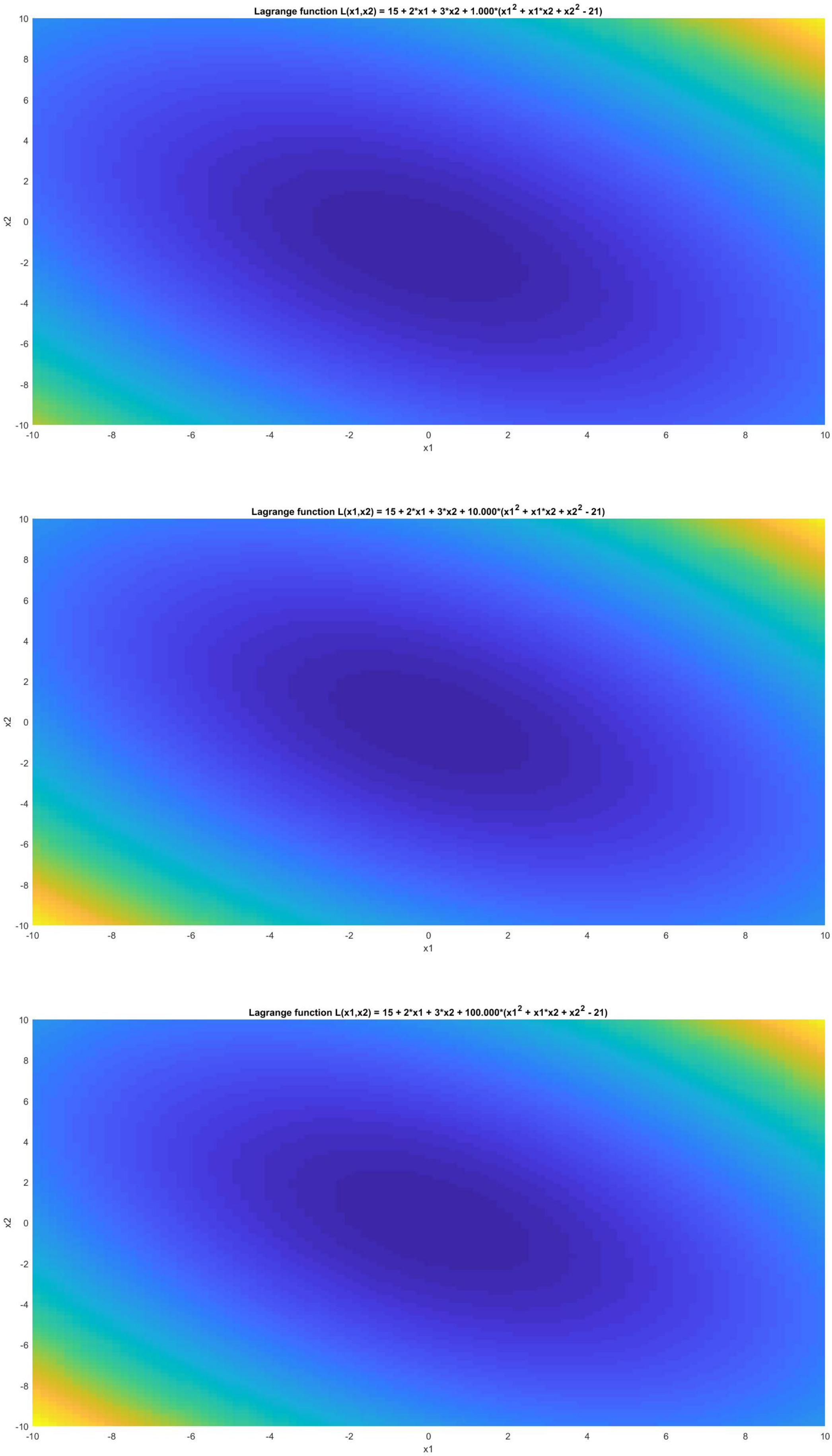
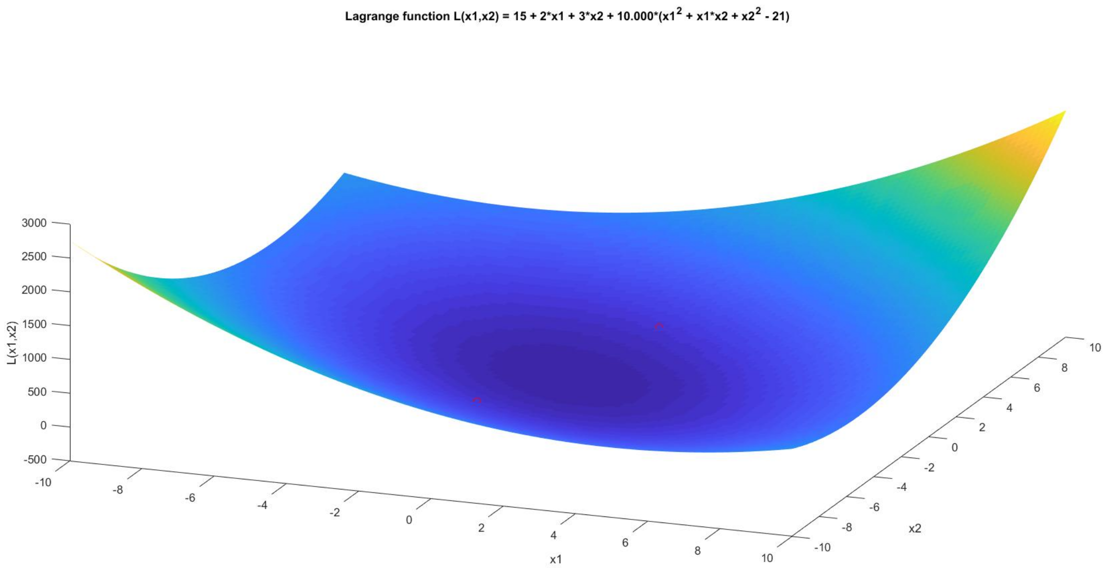


Figure 2.5: Lagrange multiplier function $L(x_1, x_2, \lambda) = f(x_1, x2) + \lambda * h(x_1, x_2)$ for sequence of λ values $[1, 10, 100]$ with vertical view on plane (x_1, x_2).

or from 3d view



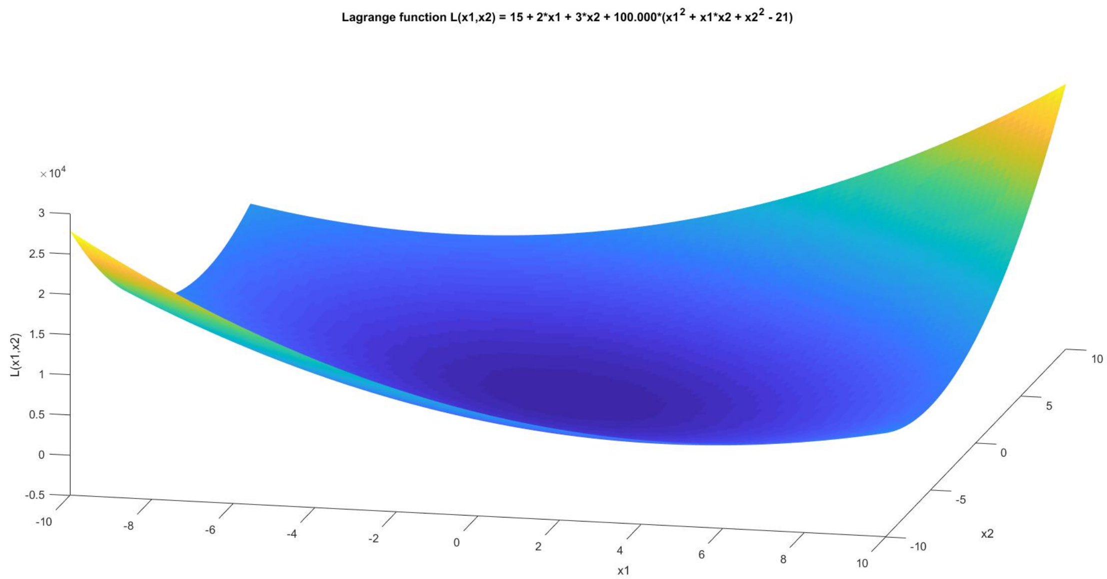


Figure 2.6: Lagrange multiplier function $L(x_1, x_2, \lambda) = f(x_1, x2) + \lambda * h(x_1, x_2)$ for sequence of λ values $[10, 100]$).

It may be seen that with change of multiplier μ value superposition of objective function $f(\vec{x})$ and constraints function $h(\vec{x})$ are switching in phase, so position of global minimum (dark blue color) is changing place too.

2.3.3 Gradient of Lagrange Multipler function

Comparing gradient of Lagrange Multipler unconstrained function to zero, step depends on solving to get points for which objective function value is optima

$$\begin{cases} \frac{\partial L}{\partial x_1} = 0 & \Leftrightarrow \quad 2 + \lambda * (2x_1 + x_2) = 0 \\ \frac{\partial L}{\partial x_2} = 0 & \Leftrightarrow \quad 3 + \lambda * (x_1 + 2x_2) = 0 \\ \frac{\partial L}{\partial \lambda} = 0 & \Leftrightarrow \quad h(x_1, x_2) = 0 \end{cases} \tag{2.20}$$

Notice that in last partial derivative over λ term is reduced to initial equality constraint $h(\vec{x})$.
From this one obtains

$$\begin{cases} \lambda = \frac{-2}{2*x_1+x_2}; \\ \lambda = \frac{-3}{x_1+2*x_2}; \end{cases} \tag{2.21}$$

Here function variables are in denominators, so it is necessary to check if optima do not happens in points corresponding to this terms. First performed is checking for denominator 1 : $2 * x_1 + x_2 = 0$ then separately for denominator 2 : $x_1 + 2 * x_2 = 0$.

Check for variables in denominator 1

If $2x_1 + x_2 = 0$ then $x_2 = -2x_1$. Inserting this to constraint function and comparing to zero (according to third partial derivative) $h(x_1, x_2 = -2x_1) = x_1^2 + x_1 * (-2x_1) + (-2x_1)^2 - 21 = 0$, after algebraic modifications $x_1^2 - 2x_1^2 + 4x_1^2 - 21 = 0$, $2x_1^2 = 21$ one obtains roots of as:

$$\begin{cases} x_1 = \pm\sqrt{7}; \\ x_2 = \mp 2\sqrt{7}; \end{cases} \tag{2.22}$$

which gives two points:

- $(\sqrt{7}, -2\sqrt{7})$ with function value $f(\sqrt{7}, -2\sqrt{7}) = 15 - 4\sqrt{7} \approx 4.41$;
- $(-\sqrt{7}, 2\sqrt{7})$ with function value $f(-\sqrt{7}, 2\sqrt{7}) = 15 + 4\sqrt{7} \approx 26$.

Check for variables in denominator 2

If $x_1 + 2x_2 = 0$ then $x_1 = -2x_2$. Inserting this to constraint function and comparing to zero (according to third partial derivative) $h(x_1 = -2x_2, x_2 =) = (-2x_2)^2 + x_2 * (-2x_2) + x_2^2 - 21 = 0$, after algebraic modifications $4x_2^2 - 2x_2^2 + x_2^2 - 21 = 0$, $3x_2^2 = 21$ one obtains roots of as:

$$\begin{cases} x_1 = \mp 2\sqrt{7}; \\ x_2 = \pm\sqrt{7}; \end{cases} \tag{2.23}$$

which gives two points:

- $(-2\sqrt{7}, \sqrt{7})$ with function value $f(-2\sqrt{7}, \sqrt{7}) = 15 + 2(-2\sqrt{7}) + 3\sqrt{7} = 15 - \sqrt{7} \approx 12$;
- $(2\sqrt{7}, -\sqrt{7})$ with function value $f(2\sqrt{7}, -\sqrt{7}) = 15 + 4\sqrt{7} - 3\sqrt{7} = 14 + \sqrt{7} \approx 18$.

Continuation of solving gradient of Lagrange function for roots

From this points 2.21 comparing terms by λ values $\frac{-3}{x_1+2*x_2} = \frac{-2}{2*x_1+x_2}$, then simplifying to $6x_1 + 3x_2 = 2x_1 + 4_{x2}$ and switching sides obtained is relation between variables $x_2 = 4x_1$. Substituting variables in equality constraint $h(x_1, 4x_1) = x_1^2 + 4x_1^2 + 16x_1^2 - 21 = 0$ then reducing $21x_1^2 = 21$ from which $x_1^2 = 1$ and finally

$$\begin{cases} x_1 = \pm 1; \\ x_2 = \pm 4; \end{cases} \tag{2.24}$$

which gives two points:

- $(1, 4)$ with function value $f(1, 4) = 15 + 2 + 3 * 4 = 29$
- $(-1, -4)$ with function value $f(-1, -4) = 15 - 2 - 12 = 1$

2.3.4 Results of Lagrange Multipler formula

Obtained points and their functions values may be grouped in table below (first point is maxima, second is minima). Both points are reaching conditions given by equality constraints, what one may check by $h(1, 4) = 0$ and $h(-1, -4) = 0$.

λ	x_1	x_2	$f_p(\vec{x}; \mu)$
-1/3	1.000	4.000	29
1/3	-1.000	-4.000	1

Table 2.2: OUTPUT : Optima points and function $f(\vec{x}, \lambda)$ values obtained with Lagrange Method for $\lambda = 1/3$

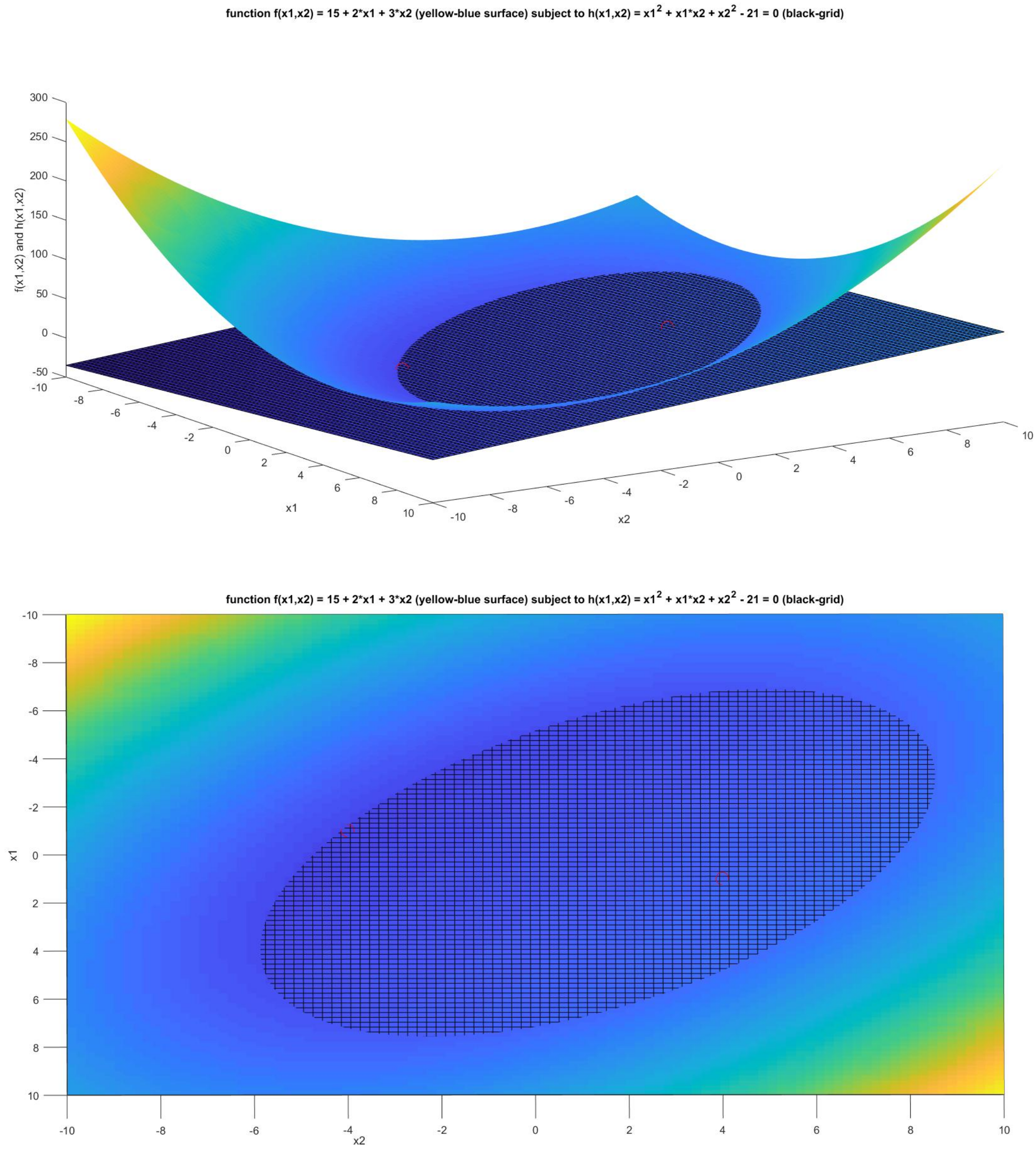


Figure 2.7: Optima points and function $f(\vec{x}, \lambda)$ values obtained with Lagrange Method for $\lambda = 1/3$, where point (1,4) with value $f(x_1, f_2) = 29$ is maxima and point (-1,-4) with value $f(x_1, f_2) = 1$ is minima.

3. Problem 1.3 - Basic GA program

3.1 Numerical solution

3.1.1 Problem Description

To finish this task, one wrote a standard genetic algorithm (GA) using (some of) the components described in Sect. 3.2.1 of the course book. Also used was the Matlab program written during the Matlab introduction with some modifications. In addition to writing the main program (FunctionOptimization.m), this task included writing Matlab functions (placed in separate M-les) for

1. initializing a population (InitializePopulation),
2. decoding a (binary) chromosome (DecodeChromosome),
3. evaluating an individual (EvaluateIndividual),
4. selecting individuals with tournament selection (TournamentSelect),
5. carrying out crossover (Cross),
6. carrying out mutations (Mutate).
7. carrying out elitism (InsertBestIndividual)

A version of each of these functions (except the one handling elitism, see below) was already implemented during the Matlab introduction. However, for this problem functions were generalized in some cases.

3.1.2 Program Metodology

To get accustomed with genetic algorithms, one had chosen ranges of parameters which were set in nested for loops for each main program run. Every each run for given parameters set was repeated 20 times for different random seed to gain average of run. From this runs were acquired parameter sets that for given problem gave averaged maximum/minimum fitness of generation and average maximum/minimum mean fitness of generation. For each of this 4 sets of parameters, runs were repeated separately to check if it is possible to gain similar scores.

3.1.3 Hardcoded Input Parameters

Input hardcoded parameters are as follows:
nCopiesBestIndividual = 2;
nVariables = 2;
nPopulations = [40 80];
nGenes = 25;
crossoverProbabilities = [0.3 0.5 0.7];
mutationProbabilities = [0.025 0.05 0.075];
tournamentSelectionParameters = [0.2 0.5 0.8];
nTournaments = [2 4 6];
variablesRange = 5.0;
nGenerations = 100;
doPlot = true;
doPrint = true;
nAveragingRuns = 3;

Where for nPopulations, crossoverProbabilities, mutationProbabilities, tournamentSelectionParameters, nTournaments runs were performed in nested for loop. Total number of runs was $2 * 3 * 3 * 3 * 3 * 20 = 3240$.

3.1.4 Program Instruction

To run program, simply open Matlab go to EvolutionaryAlgorithm directory and then run script FunctionOptimization without any changes in file due to hardcoded parameters. To see previously found best solutions one may change top settings to be: $doRunOnlyTest = false$; and $doRunOnlyForPrecalculatedSolutions = true$, what will allow to read all saved solutions and pick best/worst ones.

Typical output of program is like

*** *** *** *** *** .:: SOLUTION no. (-1), title 'test run' ::.
... parameters :
... ... population size = 80, crossover probab. = 0.5, mutation probab. = 0.025,
... ... tournament selection param. = 0.8, tournament size = 6
... results : avg/max fitness = 0.236/0.333, f(x1=-0.000, x2=-1.000) = 3.000

3.1.5 Results

From 3240 runs the best/worst and best/worst average solutions were found and rerun. Tables below shows sets of input parameters corresponding to this solutions.

Table 3.1: Worst solution (averaged over 20 runs for given set of parameters), average/max fitness (of final generation) = 0.001/0.012

parameter	parameter value
population size	40
crossover probability	0.5
mutation probability	0.025
tournament selection parameter	0.2
tournament size	2

Table 3.2: Worst average solution (averaged over 20 runs for given set of parameters), average/max fitness (of final generation) = 0.005/0.188

parameter	parameter value
population size	80
crossover probability	0.5
mutation probability	0.025
tournament selection parameter	0.2
tournament size	2

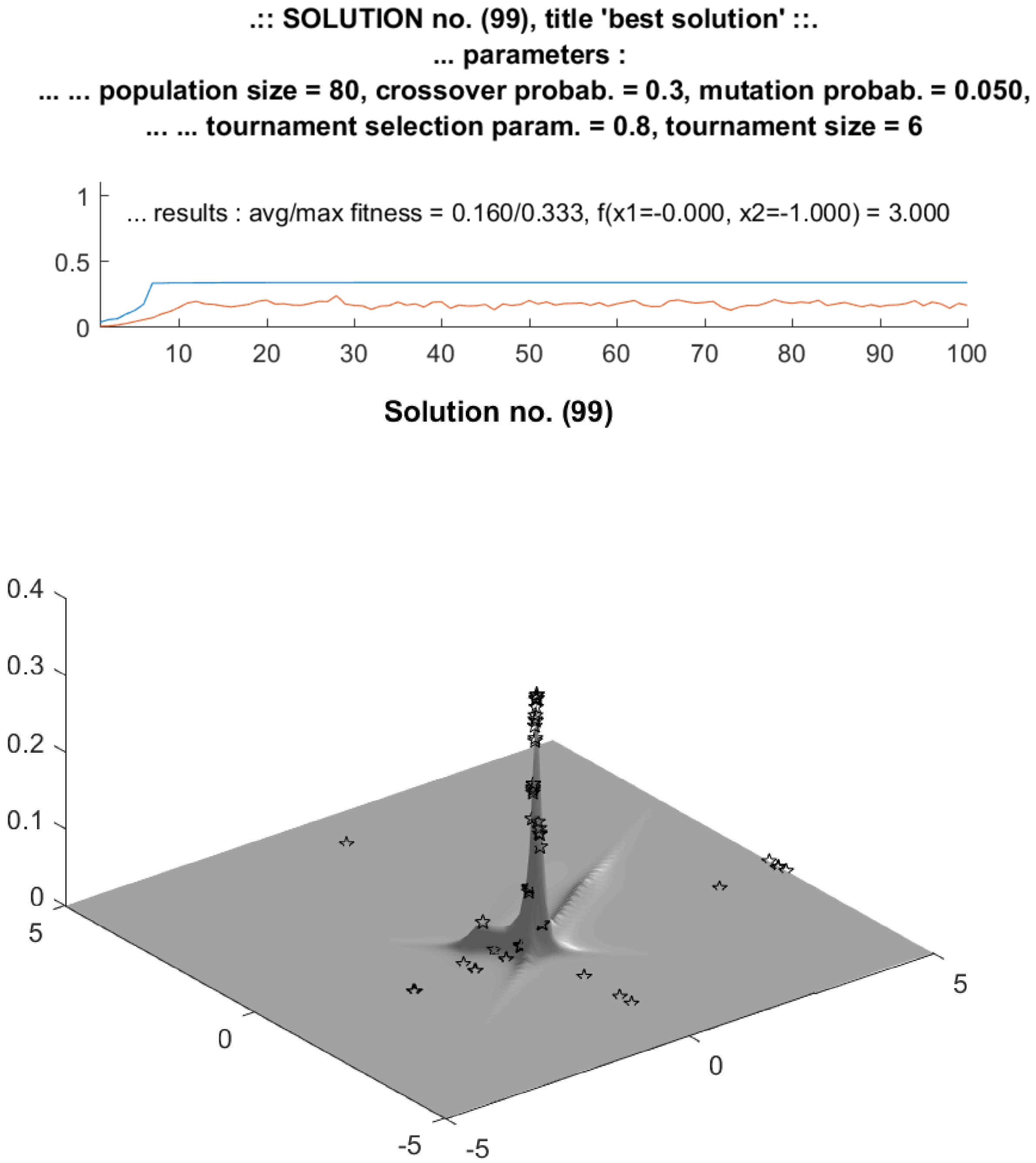
Table 3.3: Best average solution (averaged over 20 runs for given set of parameters), average/max fitness (of final generation) = 0.264/0.333

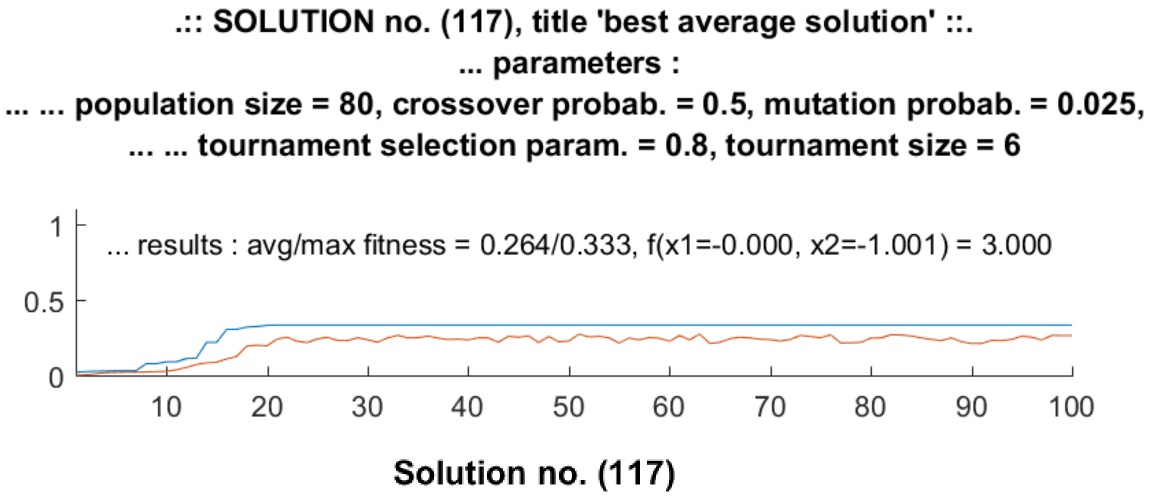
parameter	parameter value
population size	80
crossover probability	0.5
mutation probability	0.025
tournament selection parameter	0.8
tournament size	6

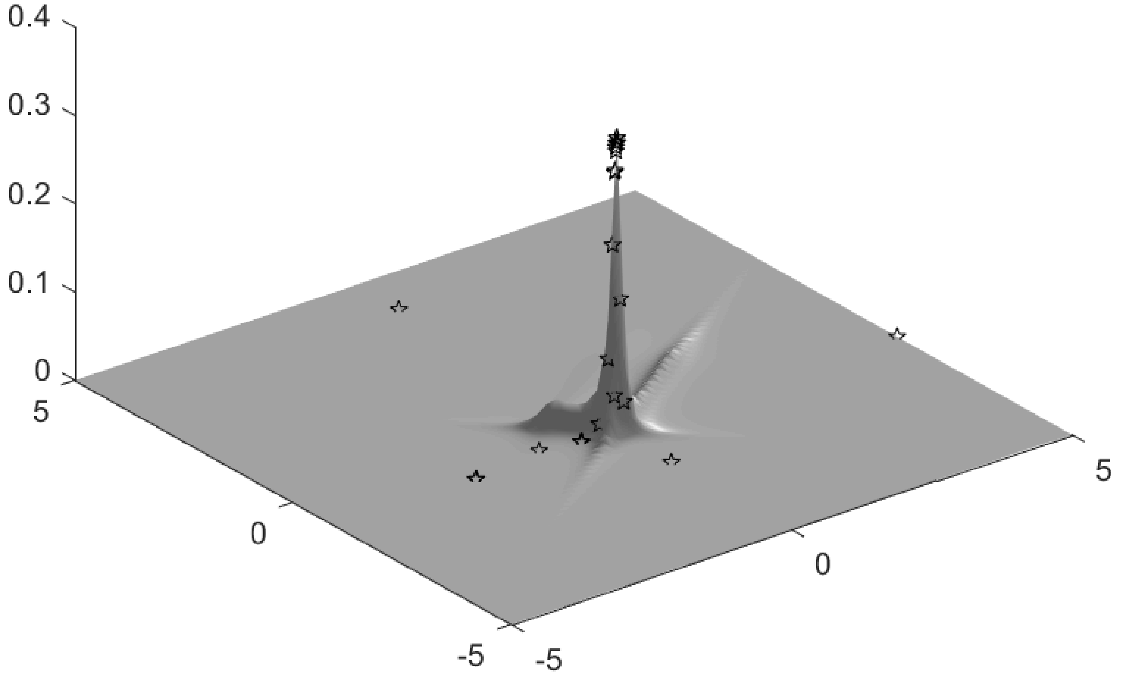
Table 3.4: Best solution (averaged over 20 runs for given set of parameters), average/max fitness (of final generation) = 0.160/0.333

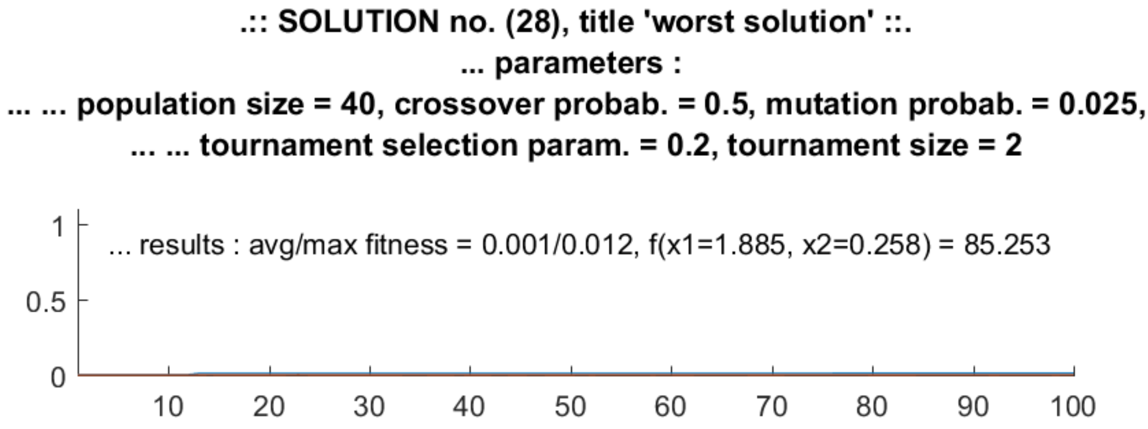
parameter	parameter value
population size	80
crossover probability	0.3
mutation probability	0.05
tournament selection parameter	0.8
tournament size	6

From best solution and best average solution one may see that stationary point of objective function $g(\vec{x})$ is (0,-1), and global minima in this point have a value 3.0.

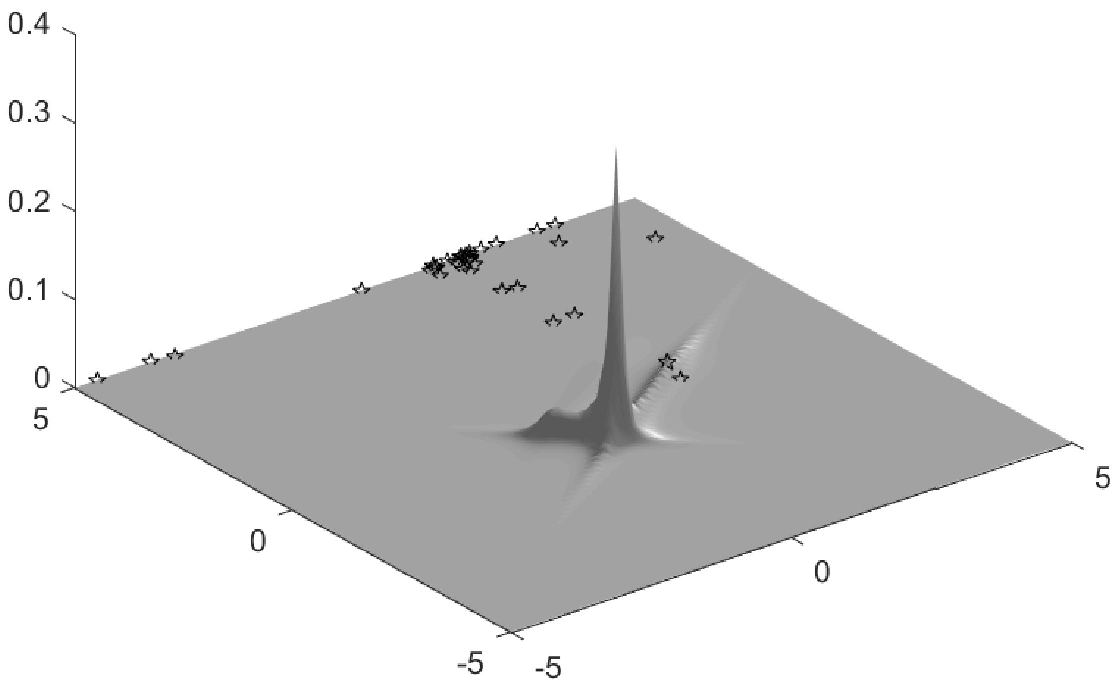

.:: SOLUTION no. (99), title 'best solution' ::.
... parameters :
... ... population size = 80, crossover probab. = 0.3, mutation probab. = 0.050,
... ... tournament selection param. = 0.8, tournament size = 6
... results : avg/max fitness = 0.160/0.333, f(x1=-0.000, x2=-1.000) = 3.000
Solution no. (99)


.:: SOLUTION no. (117), title 'best average solution' ::.
... parameters :
... ... population size = 80, crossover probab. = 0.5, mutation probab. = 0.025,
... ... tournament selection param. = 0.8, tournament size = 6
... results : avg/max fitness = 0.264/0.333, f(x1=-0.000, x2=-1.001) = 3.000
1
0.5
0
10
20
30
40
50
60
70
80
90
100
Solution no. (117)

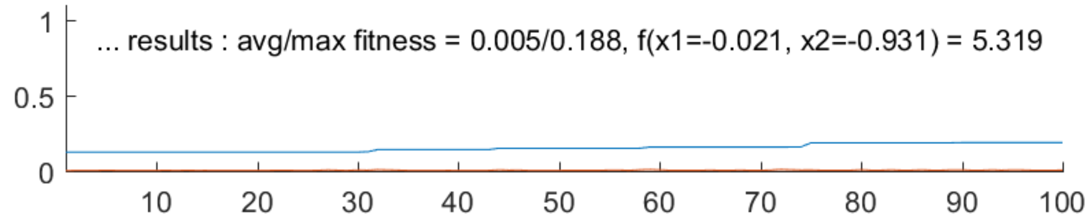

0.4
0.3
0.2
0.1
0
5
0
-5
-5
0
5



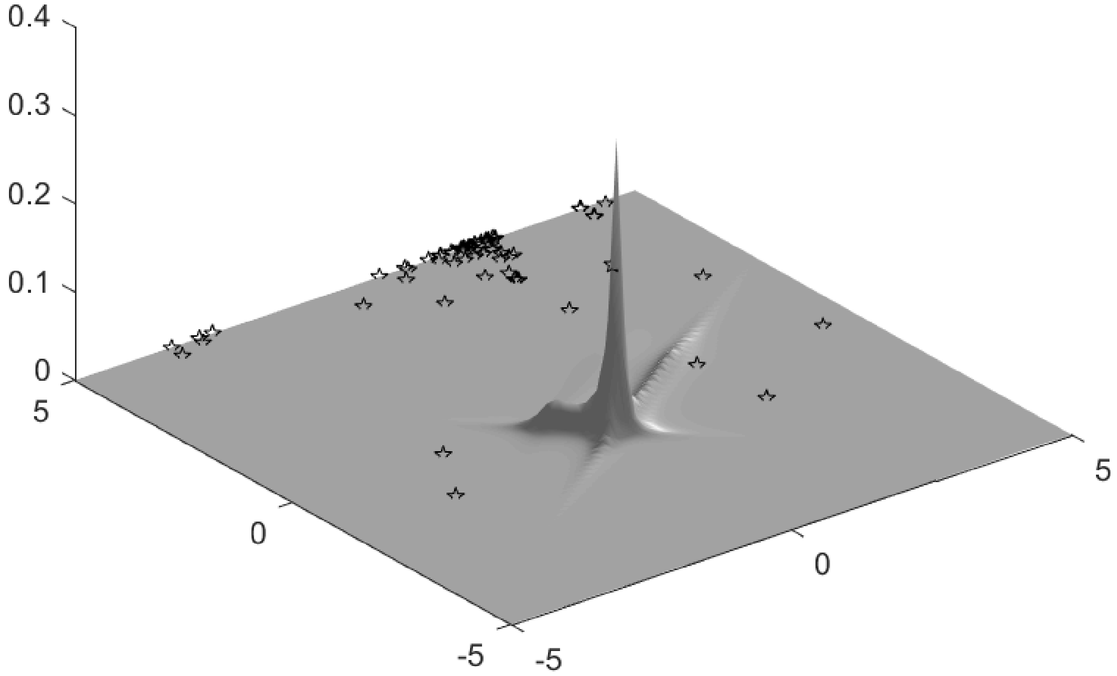
Solution no. (28)



.:: SOLUTION no. (109), title 'worst average solution' ::.
... parameters :
... ... population size = 80, crossover probab. = 0.5, mutation probab. = 0.025,
... ... tournament selection param. = 0.2, tournament size = 2



Solution no. (109)



3.1.6 Conclusions

It may be seen, that for bigger values of tournament size (6) and tournament selection parameters (0.8), with low crossover probability (0.3) and medium in sequence mutation probability (0.05), one will get best solution. Best average is achieved for similar parameters with exception of lower mutation probability (0.025) and higher crossover probability (0.3). Worst solution is when tournament input settings are low.

3.2 Analytical solution

Given objective function

$$\begin{aligned} g(\vec{x}) = (1 + (x_1 + x_2 + 1)^2 * (19 - 14x_1 + 3x_2 - 14x_2 + 6x_1x_2 + 3x_2^2)) * \\ * (30 + (2x_1 - 3x_2)^2 * (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)) \end{aligned} \tag{3.1}$$

which could be rewritten as product of two binomial functions

$$g(\vec{x}) = A(\vec{x}) * B(\vec{x}) \tag{3.2}$$

and each of this functions may be treated separately as other functions products

$$\begin{aligned} A(\vec{x}) = 1 + a(\vec{x}) * c(\vec{x}) \\ B(\vec{x}) = 30 + b(\vec{x}) * d(\vec{x}) \end{aligned} \tag{3.3}$$

Finally objective function may be formed as

$$g(\vec{x}) = (1 + a(\vec{x}) * c(\vec{x})) * (30 + b(\vec{x}) * d(\vec{x})) \tag{3.4}$$

To prove that point (0,-1) is stationary point with $g(\vec{x})$ having minima value (=3) in this point, easiest is to derive gradient of function then compare it to zero

$$\nabla g(\vec{x}) = \left(g'_{x_1}(\vec{x}), g'_{x_2}(\vec{x})\right) = \left(\frac{\partial g(\vec{x})}{\partial x_1}, \frac{\partial g(\vec{x})}{\partial x_1}\right) = 0 \tag{3.5}$$

Following function product rule in nested way first partial derivative over will be

$$g'(\vec{x}) = A'(\vec{x}) * B(\vec{x}) + A(\vec{x}) * B'(\vec{x}) \tag{3.6}$$

and for derivative terms

$$\begin{aligned} A'(\vec{x}) = a'(\vec{x}) * c(\vec{x}) + a(\vec{x}) * c'(\vec{x}) \\ B'(\vec{x}) = b'(\vec{x}) * d(\vec{x}) + b(\vec{x}) * d'(\vec{x}) \end{aligned} \tag{3.7}$$

From this our long equation partial derivatives will be

$$\begin{aligned} g'_{x_1}(\vec{x}) &= A'_{x_1}(\vec{x}) * B_{x_1}(\vec{x}) + A_{x_1}(\vec{x}) * B'_{x_1}(\vec{x}) = \\ &= (c'_{x_1}(\vec{x}) * a_{x_1}(\vec{x}) + c_{x_1}(\vec{x}) * a'_{x_1}(\vec{x})) * (b_{x_1}(\vec{x}) * d_{x_1}(\vec{x}) + 30) + \\ &+ (a_{x_1}(\vec{x}) * c_{x_1}(\vec{x}) + 1) * (b'_{x_1}(\vec{x}) * d_{x_1}(\vec{x}) + b_{x_1}(\vec{x}) * d'_{x_1}(\vec{x})) \\ g'_{x_2}(\vec{x}) &= A'_{x_2}(\vec{x}) * B_{x_2}(\vec{x}) + A_{x_2}(\vec{x}) * B'_{x_2}(\vec{x}) = \\ &= (c'_{x_2}(\vec{x}) * a_{x_2}(\vec{x}) + c_{x_2}(\vec{x}) * a'_{x_2}(\vec{x})) * (b_{x_2}(\vec{x}) * d_{x_2}(\vec{x}) + 30) + \\ &+ (a_{x_2}(\vec{x}) * c_{x_2}(\vec{x}) + 1) * (b'_{x_2}(\vec{x}) * d_{x_2}(\vec{x}) + b_{x_2}(\vec{x}) * d'_{x_2}(\vec{x})) \end{aligned} \tag{3.8}$$

After changing symbolical expressions of functions

$$\begin{aligned}
g'_{x_1}(\vec{x}) = {} & ((6 * x_1 + 6 * x_2 - 14) * (x_1 + x_2 + 1)^2 + \\
& + (2 * x_1 + 2 * x_2 + 2) * (3 * x_1^2 + 6 * x_1 * x_2 - 14 * x_1 + 3 * x_2^2 - 14 * x_2 + 19)) * \\
& * ((2 * x_1 - 3 * x_2)^2 * (12 * x_1^2 - 36 * x_1 * x_2 - 32 * x_1 + 27 * x_2^2 + 48 * x_2 + 18) + 30) + \\
& + ((x_1 + x_2 + 1)^2 * (3 * x_1^2 + 6 * x_1 * x_2 - 14 * x_1 + 3 * x_2^2 - 14 * x_2 + 19) + 1) * \\
& * ((8 * x_1 - 12 * x_2) * (12 * x_1^2 - 36 * x_1 * x_2 - 32 * x_1 + 27 * x_2^2 + 48 * x_2 + 18) - \\
& + (2 * x_1 - 3 * x_2)^2 * (36 * x_2 - 24 * x_1 + 32)) \\
g'_{x_2}(\vec{x}) = {} & ((6 * x_1 + 6 * x_2 - 14) * (x_1 + x_2 + 1)^2 + (2 * x_1 + 2 * x_2 + 2) * \\
& * (3 * x_1^2 + 6 * x_1 * x_2 - 14 * x_1 + 3 * x_2^2 - 14 * x_2 + 19)) * \\
& * ((2 * x_1 - 3 * x_2)^2 * (12 * x_1^2 - 36 * x_1 * x_2 - 32 * x_1 + 27 * x_2^2 + 48 * x_2 + 18) + 30) - \\
& + ((x_1 + x_2 + 1)^2 * (3 * x_1^2 + 6 * x_1 * x_2 - 14 * x_1 + 3 * x_2^2 - 14 * x_2 + 19) + 1) * \\
& * ((12 * x_1 - 18 * x_2) * (12 * x_1^2 - 36 * x_1 * x_2 - 32 * x_1 + 27 * x_2^2 + 48 * x_2 + 18) - \\
& + (2 * x_1 - 3 * x_2)^2 * (54 * x_2 - 36 * x_1 + 48))
\end{aligned} \tag{3.9}$$

Substituting stationary point (0,-1) for roots of equation

$$\begin{aligned}
g'_{x_1}(0,-1) = {} & ((6 * 0 + 6 * (-1) - 14) * (0 + (-1) + 1)^2 + \\
& + (2 * 0 + 2 * (-1) + 2) * (3 * 0^2 + 6 * 0 * (-1) - 14 * 0 + 3 * (-1)^2 - 14 * (-1) + 19)) * \\
& * ((2 * 0 - 3 * (-1))^2 * (12 * 0^2 - 36 * 0 * (-1) - 32 * 0 + 27 * (-1)^2 + 48 * (-1) + 18) + 30) + \\
& + ((0 + (-1) + 1)^2 * (3 * 0^2 + 6 * 0 * (-1) - 14 * 0 + 3 * (-1)^2 - 14 * (-1) + 19) + 1) * \\
& * ((8 * 0 - 12 * (-1)) * (12 * 0^2 - 36 * 0 * (-1) - 32 * 0 + 27 * (-1)^2 + 48 * (-1) + 18) - \\
& + (2 * 0 - 3 * (-1))^2 * (36 * (-1) - 24 * 0 + 32)) \\
g'_{x_2}(0,-1) = {} & ((6 * 0 + 6 * (-1) - 14) * (0 + (-1) + 1)^2 + (2 * 0 + 2 * (-1) + 2) * \\
& * (3 * 0^2 + 6 * 0 * (-1) - 14 * 0 + 3 * (-1)^2 - 14 * (-1) + 19)) * \\
& * ((2 * 0 - 3 * (-1))^2 * (12 * 0^2 - 36 * 0 * (-1) - 32 * 0 + 27 * (-1)^2 + 48 * (-1) + 18) + 30) - \\
& + ((0 + (-1) + 1)^2 * (3 * 0^2 + 6 * 0 * (-1) - 14 * 0 + 3 * (-1)^2 - 14 * (-1) + 19) + 1) * \\
& * ((12 * 0 - 18 * (-1)) * (12 * 0^2 - 36 * 0 * (-1) - 32 * 0 + 27 * (-1)^2 + 48 * (-1) + 18) - \\
& + (2 * 0 - 3 * (-1))^2 * (54 * (-1) - 36 * 0 + 48))
\end{aligned} \tag{3.10}$$

Then simplifying one finds out that for given roots partial derivatives are equal zero

$$\begin{aligned}
g'_{x_1}(0,-1) = 0 \\
g'_{x_2}(0,-1) = 0
\end{aligned} \tag{3.11}$$

What is proving that point (0,-1) is stationary point of objective function $g(x_1, x_2)$.