OPTIMIZATION

DETERMINISTIC OPTIMIZATION - NON-LINEAR CONSTRAINED OPTIMIZATION USING THE PENALTY METHOD

Author: Lawrence Adu-Gyamfi (1484610)
Date: 10/02/2019

Table of Contents

1. INTRODUCTION

In Mathematical optimization we seek to determine the extreme values (minimum, maximum) of a given function in a stipulated domain. Constrained optimization is a class of optimization problems in which we seek to solve the optimization problem subject to some equality and/or inequality constraints.

The assignment problem is to optimise the following non linear functions subjected to a set of constraints and using the penalty method. Below are the functions to be optimized.

The following table shows each problem and a 3D plot of the problem together with the constraints.

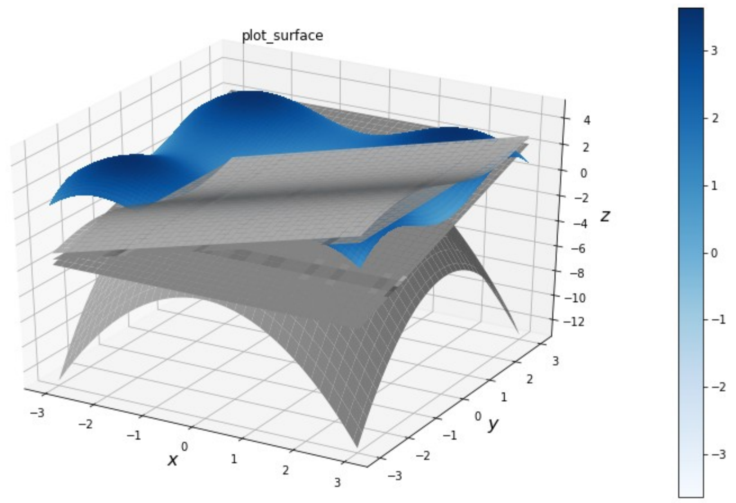
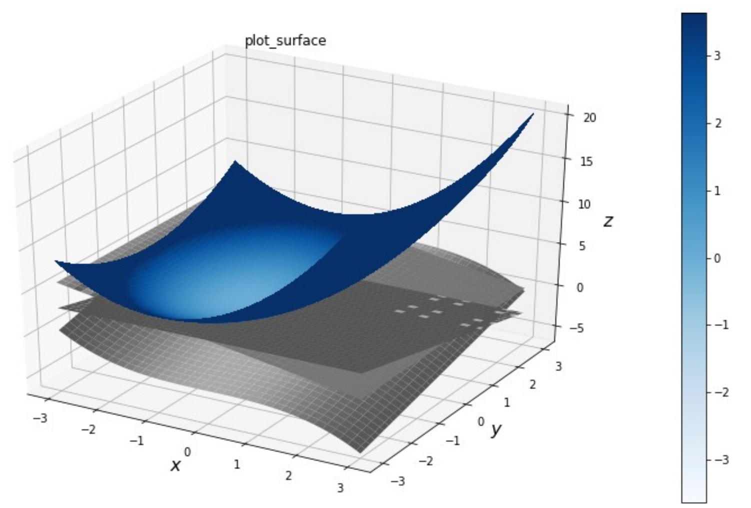
$\min \; x\sin(x) + y\sin(y)$ $\text{s.t.} \begin{cases} x \geq \frac{1}{3} \\ y \geq \frac{3}{4} \\ x - \sin(y) \geq 0 \\ x^2 + y^2 \leq 5 \end{cases}$	
$\min \; (x+1)^2 + \frac{1}{2}y^2$ $\text{s.t.} \begin{cases} x \leq 3 \\ y \geq 0 \\ \frac{1}{8}x^3 - y \geq 0 \end{cases}$	

Table 1-1:Problem-specification showing constraints

2. METHODOLOGY

This section explains the methodology that has been implemented in solving the optimization problem. A brief description of the penalty method is given below.

2.1. PENALTY METHOD ref [2]

In utilising the penalty method to solve a constrained optimization problem, we seek to approximate the solution to the problem (objective function) using an unconstrained one and then applying standard line search techniqes to obtain the solution(ref const_opt). Line search tmethods are iterative techniques that terminate in their attempt to approximate the solution to a problem when a predefined stopping criteria is satisfied.

A term in added to the objective function that assigned a large cost for violation of the constraints of the problem.
For instance, considering the following optimization problem:

$$\text{Minimize}\{f(\mathbf{x}) : \mathbf{x} \in S\}$$

The penalty method solves this by replacing it with an unconstrained problem such as :

$$\text{Minimize}\{f(\mathbf{x}) + cP(\mathbf{x})\}$$

where:
c – is a positive constant
P – is continous and smooth function made up the effect of violations of the constraints of the chosen x based on the penalty function used.

The unconstrained function is termed as the augmented objective function. If value of c is made suffificently large, the penalty term will assign a huge cost to any violations of the constraints. As a result minimizing the augmented objective function eventually leads to a feasible solution for the original objective function.

Below is the algorithm for solving the constrained optimization problem using the penalty method as used in the implementation presented in this report.

<u>Initialisation</u>
select:
growth parameter n>1
stopping parameter e > 0
initial value for c0

choose a starting point for x0 that violates at least one constraint and formulate augmented objective function, let k =1

Iterative step

starting from x(k-1), use unconstrained search technique to find the point that minimizes theta. Call it xk and deternine which constraints are violated at this point.

Stopping rule:

- if the distance between x(k-1) and xk is smaller than e x(k-1) – xk < e or the difference between the results of two successive objective function computations is smaller than e, stop with xk as etsimate for optimal solution
- else:
- put ck = n*ck-1, formulate new theta based on which constraints are violated at xk and put k=k+1 and repeat iterative step

2.2. IMPLEMENTATION

The solution to the above stated problems have been implemented in python(ver 3.6). The methodology of implementation is similar in both cases with only the specific functions and constraints changing. Also the starting positions have been alterned to suit each problem together with the penalty values and growth terms.
Consequently two(2) files are provided, with 1 for each problem.(deterministic_1.py and deterministic_2.py)

A run of each program solves the corresponding problem and generates the corresponding covergence graph.

3. RESULTS

This section presents the results showing the solutions of both problems.

3.1. SOLUTION TO QUESTION 1

The figure below shows the results of the optimization of the first problem based on the constraints provided. The feasible region is shown with the red star highlighting the minimum of the function in the region.

The convergence of the solution over the iterations is provided as well showing no change in the value of the function after a number of iteration.

The solution has been obtained with a growth parameter of 0.000001 and a C value of 40000.

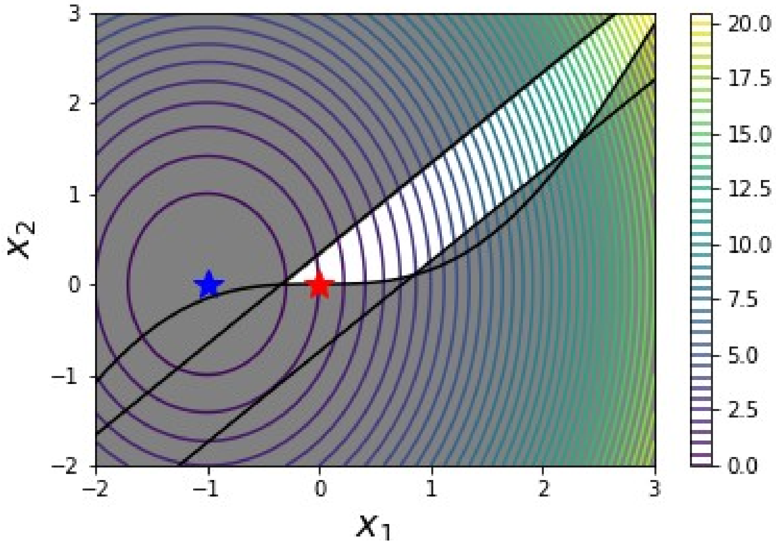


Figure 2-2:Solution to xsinx + ysiny based on constraints

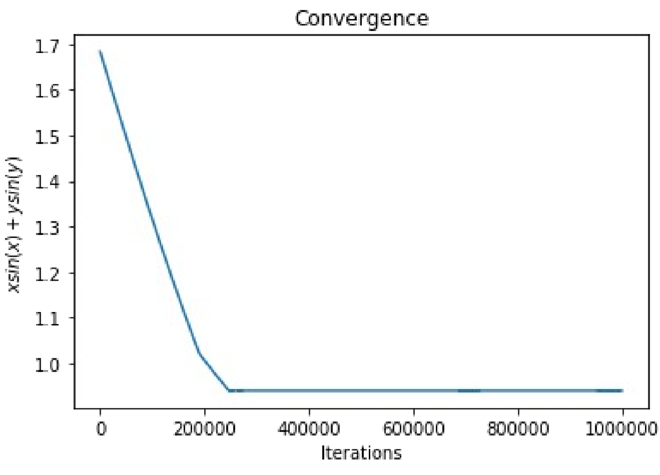


Figure 2-3:Solution onvergence

3.2. SOLUTION TO QUESTION 2

The figure below shows the results of the optimization of the second problem based on the constraints provided. The feasible region is shown with the red star highlighting the minimum of the function in the region.

The convergence of the solution over the iterations is provided as well showing no change in the value of the function after a number of iteration.

The solution has been obtained with a growth parameter of 0.00001 and a C value of 1000000.

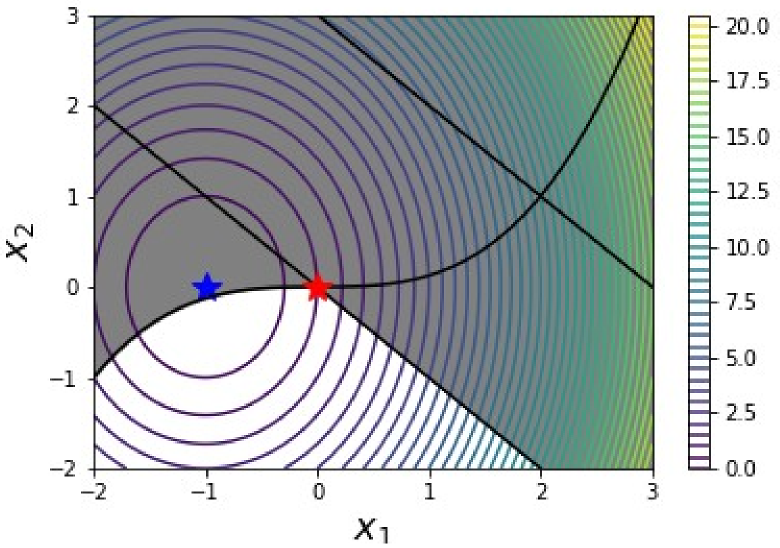


Figure 2-4:Solution to $(x+1)^2 + 1/2*(y^2)$ based on constraints

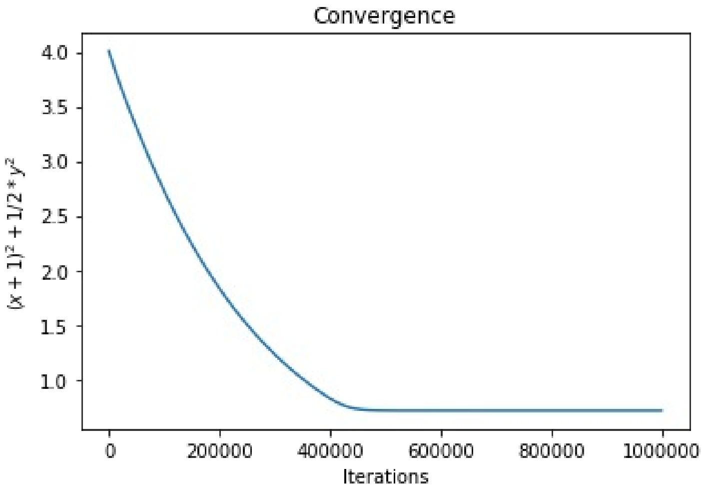


Figure 2-5:Solution onvergence

4. CONCLUSION

As part of this exercise, a python program has been developed to implement the solution of two (2) constrained optimization problems. The penalty method has been used to translate the constrained problem to an unconstrained one before solving.
Choosing a small value for the penalty was noticed to either slow the execution of the program or result in errors. This highlights the importance of choosing a large value for C in order to assign large penalties for solutions that violate the constraints.

4.1. REFERENCES

The following references have been used in one way or the other in preparation of this report. They have been referenced as and when applicable.

1. Numerical Mathematics, *Alfio Quarteroni*, *Riccardo Sacco*, *Fausto Saleri*
2. Penalty and Barrier Methods for Constrained Optimization *(Supplementary Transparency provided by Lecturer)*
3. Numerical Python, *Robert Johansson*