Home Problem 1

Anton Karlsson, 930225-1179, antkarls@student.chalmers.se

September 19, 2017

FFR105, Stochastic Optimization Algorithms
Chalmers Tekniska Hgskola

Problem 1.1)

The function f_p for the given problem is defined as follows:

$$f_p(\boldsymbol{x};\mu) = \begin{cases} (x_1-1)^2 + 2(x_2-2)^2 & \text{if } x_1^2 + x_2^2 - 1 \leq 0 \\ (x_1-1)^2 + 2(x_2-2)^2 + \mu\left(x_1^2 + x_2^2 - 1\right)^2 & \text{otherwise} \end{cases}$$

Finding the gradient is then a matter of simple differentiation and it is calculated by taking the derivative with respect to each variable.

$$\nabla f_p(\boldsymbol{x};\mu) = \begin{cases} \begin{bmatrix} 2(x_1-1) \\ 4(x_2-2) \end{bmatrix} & \text{if } x_1^2 + x_2^2 - 1 \leq 0 \\ \\ \begin{bmatrix} 2(x_1-1) + 4\mu x_1\left(x_1^2 + x_2^2 - 1\right) \\ 4(x_2-2) + 4\mu x_2\left(x_1^2 + x_2^2 - 1\right) \end{bmatrix} & \text{otherwise} \end{cases}$$

Because the unconstrained function is a sum of two squares, it can never take negative values, therefore the obvious stationary point $(1,2)^T$ where the functionvalue is zero is one local minimum of the unconstrained function. It can further be noted that the eigenvalues of the Hessian are 2 and 4 (positive definite), which means that the function is convex and any local minimum is also a global minimum. The starting point for the gradient descent will therefore be $(1,2)^T$.
To solve the optimization problem the method of gradient descent was used. To run the program implementing this, simply call the script PenaltyMethod.m included in the handed in .zip folder. This program creates and prints a table with stationary points found for different values of the penalty parameter, μ. The program will call the function GradientDescent.m which will in turn continue to iterate using the gradient descent method until the modulus of the gradient at the current iteration point is smaller than a given threshold (10^{-6}), or if 10^5 iterations has been made, in which case an error message is printed to the Matlab console.
The table generated by the program is the following :

μ	x$_1$	x$_2$
1	0.434	1.210
10	0.331	0.996
100	0.314	0.955
1000	0.312	0.951

To check if the series is converging, one can examine the fraction of two adjacent values, and if this quotient is approaching 1 then there is convergence. For x_1 the fractions are 0.763, 0.949 and 0.994, and for x_2 the equiavalent fractions are 0.8231, 0.9588 and 0.9958. Both these series are approaching 1 and it seems like the stationary point found by the program is converging for larger μ. The found point lies on the boundary, which is reasonable since the function is a bowl

Problem 1.2a)

"Analytical method"

Following the procedure outlined at page 29 in the course textbook we start by finding the stationary points of $f(\boldsymbol{x})$ in the interior of $\boldsymbol{S}$. We do this by finding the gradient and see where it's zero. The resulting equations are as follows

$$\begin{cases} \frac{\partial f}{\partial x_1} & = 8x_1 - x_2 = 0 \\ \frac{\partial f}{\partial x_2} & = -x_1 + 8x_2 - 6 = 0 \end{cases}$$

Solving this simple line intersection yields one solution $P_1 = \frac{2}{21}(1,8)^T$, which belongs to the interior of $\boldsymbol{S}$, and is therefore our first stationary point to consider.[1]
Next we restrict $f(\boldsymbol{x})$ to the boundary $\partial\boldsymbol{S}$. We start with $x_1 = x_2$ where $x_1, x_2 \in [0,1]$. The objective function now reduces to a one-dimensional function and the equation from setting the derivative to zero is:

$$14x - 6 = 0$$

The solution is our second stationary point $P_2 = \frac{3}{7}(1,1)^T$ to evaluate. Next boundary is when $x_1 = 0$ and $x_2 \in [0,1]$. Again, the objective function reduces to a one-dimensional function and the equation from setting the derivative to zero is

$$8x_2 - 6 = 0$$

The solution $P_3 = (0, \frac{3}{4})^T$ is our next point to consider. The case $x_2 = 1$ and $x_1 \in [0,1]$ is checked next, just like the cases above, the function reduces to a one dimension, and the resulting equations is

$$8x_1 - 1 = 0$$

which has the solution $P_4 = (\frac{1}{8}, 1)^T$.
Finally the corners are to be included, these are $P_5 = (0,0)^T$, $P_6 = (0,1)^T$ and $P_7 = (1,1)^T$. The object function evaluated at all points of interest is outlined in the table below:

Point	Function value
$\text{P}_1 = \frac{2}{21}(1,8)^T$	-2.29
$\text{P}_2 = \frac{3}{7}(1,1)^T$	-1.28
$\text{P}_3 = (0, \frac{3}{4})^T$	-2.25
$\text{P}_4 = (\frac{1}{8}, 1)^T$	-2.06
$\text{P}_5 = (0,0)^T$	0
$\text{P}_6 = (0,1)^T$	-2
$\text{P}_7 = (1,1)^T$	1

The conclusion is, as expected, that the minimum of the objective function, -2.29, occurs at $\frac{2}{21}(1,8)^T$.

[1]By finding the eigenvalues of the Hessian, which in this case are all positive, one can see that the function is convex, and therefore any local minimum is also a global minimum. Since we already found one local minimum which happens to be a feasible point, we know this is going to be our global minimum, even for the constrained problem.

Problem 1.2b)

"Lagrange multiplier method"

Noting that at a minimum, the gradient vector of the objective funtion and the equality constraint has to be parallel we can form the function

$$L(x_1, x_2, \lambda) = 15 + 2x_1 + 3x_2 + \lambda(x_1^2 + x_1x_2 + x_2^2 - 21)$$

and find the stationary points of this to find the minimum value of the constrained function. Setting the gradient of L to zero yields the following equation system:

$$\begin{cases} \frac{\partial L}{\partial x_1} = 2 + \lambda(2x_1 + x_2) = 0 \\ \frac{\partial L}{\partial x_2} = 3 + \lambda(x_1 + 2x_2) = 0 \\ \frac{\partial L}{\partial \lambda} = x_1^2 + x_1x_2 + x_2^2 - 21 = 0 \end{cases}$$

The case when either x_1 or x_2 is zero will be investigated separately, so in the calculations below we can assume that neither of the variables are zero. The equation system above can be solved for λ with some simple algebra, giving two expressions for λ.

$$2\frac{\partial L}{\partial x_2} - \frac{\partial L}{\partial x_1} = 4 + 3\lambda x_2 = 0 \;\Rightarrow\; \lambda = -\frac{4}{3x_2}$$

and

$$2\frac{\partial L}{\partial x_1} - \frac{\partial L}{\partial x_1} = 1 + 3\lambda x_1 = 0 \;\Rightarrow\; \lambda = -\frac{1}{3x_1}$$

Setting the two expressions for λ equal to eachother we find that $4x_1 = x_2$. Used in the equality constraint we can see that $x_1^2 = 1$ and thus the stationary points are $\pm(1, 4)^T$ with the corresponding function values 29 and 1.
The special cases when one of the variables is zero (both can't be zero at the same time due to the constraint), we can easily get the points $(0, \pm\sqrt{21})^T$ and $(\pm\sqrt{21}, 0)^T$ from the constraint equation. The resulting function values for these four points are in order 22.93, 7.06, 20.29, 9.71, all of which are larger than 1.
In conclusion the constrained objective function has a minimum at $(-1, -4)^T$ where it takes the value 1.

Problem 1.3)

Finding the global minimum

Letting the GA run for 200 generations, with $N = 100$, $P_{cross} = 0.8$, $P_{mut} = 2/m = 0.033$, $P_{tour} = 0.75$, 30 genes per variable and tournament size 2, the majority of runs finds a minimum at $(0.000, -1.000)^T$ with a function-value of 3.000.
The function is recognized as the GoldStein-Price function with known minimum of 3 at precisely the point found. We can assume the program has worked as intented. This is further reassured from the visualizations provided by the program, cleary showing the global minimum and that the individuals are able to find it after about 50 generations for a typical run.

Trying different parameter sets

A few more sets of parameters were tested as suggested in the problem description. The deafault parameter setup is the one described above, but for each table I have varied a different parameter to investigate its effect on the algoritm. For each parameter value a table with the average function minimum over 30 runs as well as the standard deviation is presented. It should be noted that once and a while the algorithm got stuck in local optima which affected the average quite a bit. However, there is still valuable conclusions to be drawn from the results.
Table 1 shows how a varying population size affects the performance of the evolutionary algorithm. Not surprisingly a larger population size increases the performance because there is simply more genetic material in the mix to be evaluated. An increased population size is however more computationally demanding and takes longer to evaluate. For this specific problem a population size of 100 consistently converged to global minimum without taking too long time to evaluate. Of course this is very case-dependent and a problem in which the evaluation of individuals is more complex, a smaller population size is probably needed.
Next I tried varying the crossover probability, as shown in table 2. This parameter had seemingly little impact on the algorithm where all the tried values roughly performed the same. This could be because the population size was large compared to the complexity of the individuals, probably some of the 100 initialized individuals already were fit enough to allow selection and mutation alone to nudge it to the global minimum. Since 30 genes were used per variable, but only 4 decimal places were represented in the table, alot of genes were of little impact to the decoded chromosome. Had messy encoding been used, the effect of crossover probably would have been more noticable.
Finally the mutation rate was investigated and the results are shown in table 3. In the case were no mutation occured, the algorithm performed very bad. This is because no new genetic material was introduced to the population. If any gene is lost during selection it has no way to reappear in the population, causing the algorithm to converge prematurely. In the case were any given gene always mutated the algorithm performed equally bad, if not slightly worse, since any progress made by selection and crossover was practically inversed by mutation. A good mutation rate seem to be some integer multiple of the inverse chromosome length, in this case 2 ,which is also the case found in the course text book, worked well.

Results from different parameter sets

Table 1: How varying population size affects the EA. Other parameters are specified on previous page.

N	min $(g(x_1, x_2))$	
	Avg.	S.D.
10	58.2369	65.8996
25	16.7557	30.9217
50	5.7043	14.7899
100	3.0033	0.0162
200	3.0004	0.0015

Table 2: How varying crossover probability affects the EA. Other parameters are specified on previous page.

P_{cross}	$\min(g(x_1, x_2))$	
	Avg.	S.D.
0	3.0002	0.0011
0.4	3.0008	0.0019
0.6	3.0001	0.0002
0.8	3.0007	0.0017
1	3.0523	0.2821

Table 3: How varying mutation rate affects the EA. Other parameters are specified on previous page.

P_{mut}	$\min(g(x_1, x_2))$	
	Avg.	S.D.
0	84.8859	101.4551
1/2m	28.5356	35.7393
1/m	12.1504	24.9600
2/m	3.0022	0.0062
10/m	3.1166	0.1501
1	92.7011	111.9033

1.2c)

Proof that $(0,-1)^T$ is a stationary point
The minimum found by the EA was $(0,-1)^T$. To prove analytically that this is a stationary point we have to compute the gradient evaluated at this point and prove it is equal to zero. This can be somewhat simplified by inserting the value of the variable we are not taking the derivate in respect to, before taking the partial derivative, i.e when computing the partial derivative with respect to x_1 we insert the value of x_2 in the equation. Since $x_1 = 0$ I will begin with $\partial g/\partial x_2$:

$$g(0,x_2) = \left(1 + (x_2+1)^2(10 - 14x_2 + 3x_2^2)\right)\left(30 + 9x_2^2(18 + 48x_2 + 27x_2^2)\right)$$

we then have

$$\begin{aligned}\frac{\partial g}{\partial x_2}(0,x_2) = &\left(2(x_2+1)(10 - 14x_2 + 3x_2^2) + (x_2+1)^2(-14 + 6x_2)\right)\left(30 + 9x_2^2(18 + 48x_2 + 27x_2^2)\right) + \\ &\left(1 + (x_2+1)^2(10 - 14x_2 + 3x_2^2)\right)\left(18x_2(18 + 48x_2 + 27x_2^2) + 9x_2^2(48 + 54x_2)\right)\end{aligned}$$

which then evaluates to

$$\begin{aligned}\frac{\partial g}{\partial x_2}(0,-1) = &\left(2\cancelto{0}{(-1+1)}(10+14+3) + \cancelto{0}{(-1+1)^2}(-14-1)\right)(30 + 9(18 - 48 + 27)) \\ &+ \left(1 + \cancelto{0}{(-1+1)^2}(10+14+3)\right)(-18(18 - 48 + 27) + 9(48 - 54)) = 0\end{aligned}$$

Doing the equivalent calculations for x_1 yields

$$g(x_1,-1) = \left(1 + x_1^2(36 - 20x_1 + 3x_1^2)\right)\left(30 + (2x_1+3)^2(-3 + 4x_1 + 12x_1^2)\right)$$

And the partial derivative then is

$$\begin{aligned}\frac{\partial g}{\partial x_2}(x_1,-1) = &\left(2x_1(36 - 20x_1 + 3x_1^2) + x_1^2(-20 + 6x_1)\right)\left(30 + (2x_1+3)^2(-3 + 4x_1 + 12x_1^2)\right) + \\ &\left(1 + x_1^2(36 - 20x_1 + 3x_1^2)\right)\left(4(2x_1+3)(-3 + 4x_1 + 12x_1^2) + (2x_1+3)^2(4 + 24x_1)\right)\end{aligned}$$

which at the given point of interest evaluates to

$$\frac{\partial g}{\partial x_2}(0,-1) = 4\cdot 3\cdot(-3) + 3^2\cdot 4 = 0$$

Since the gradient of g vanishes at $(0,-1)^T$ it is evident that it really is a stationary point.