Report of FFR105 Homework Problem 1

Siyu Hu, gushusii@student.gu.se, ID: 199509103702

Temporary ID on Ladok: 19950910-T805

1 HP1.1 Penalty Method

1.1 Define function f_p

$$f_p = f(x_1, x_2) + \mu * \max\{0, (g(x_1, x_2))\}^2$$

with

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

1.2 Compute $\nabla f_p(x_1, x_2; \mu)$

if the constrain is not fulfilled, then :

$$\partial f_p/\partial x_1 = 2(x_1 - 1) + \mu \times 2(x_1^2 + x_2^2 - 1) \times (2x_1)$$

$$\partial f_p/\partial x_2 = 4(x_2 - 2) + \mu \times 2(x_1^2 + x_2^2 - 1) \times (2x_2)$$

otherwise :

$$\partial f_p/\partial x_1 = 2(x_1 - 1)$$
$$\partial f_p/\partial x_2 = 4(x_2 - 2)$$

1.3 Find unconstrained minimum of $f(x_1, x_2)$

To fine the minimum of $f(x_1, x_2)$, we should set the $\nabla f(x_1, x_2) = 0 \Leftrightarrow \partial f/\partial x_1 = 2(x_1 - 1) = 0$ and $\partial f/\partial x_2 = 4(x_2 - 2) = 0$. Solve these two equations, we get $(x_1, x_2) = (1, 2)$, this is the start point for gradient descent.

1.4 Write a Matlab program

(details in the .zip)

1.5 Chose a list of μ and run the program in the sequence

Set $\eta = 0.0001$, $T = 10^{-6}$, when μ varies, the output of optimum points shows as below:

x_1^*	x_2^*	μ
0.4338	1.2102	1
0.3314	0.9955	10
0.3137	0.9553	100
0.3118	0.9507	1000

Table 1: Optimum points(x_1^*, x_2^*) with the sequence of μ values

Observation from Table 1: As μ increases from 1 to 1000, the values of x_1^* and x_2^* show a decreasing trend and gradually become more stable, indicating that the sequence of points (x_1^*, x_2^*) tends to converge to a specific region in the solution space as μ becomes larger. Meanwhile, for $\mu \geq 100$, the change in the optimal points becomes minimal. This suggests that the sequence has reached a relatively stable status, and approached the convergence state.

2 HP 1.2 Constrained optimization

To optimize the objective function $f(x_1, x_2)$ with the inequality constrain $g(x_1, x_2) = x_1^2 + x_2^2 - 4 \leq 0$, we need to both check the interior and boundary of the constrain.

2.1 Check the interior

Find the critical points inside the set without considering the constraint.

$$\frac{\partial f}{\partial x_1} = 8x_1 = 0 \quad \Rightarrow \quad x_1 = 0$$

$$\frac{\partial f}{\partial x_2} = 6x_2^2 = 0 \quad \Rightarrow \quad x_2 = 0$$

So the critical point is: $(x_1, x_2) = (0, 0)$.
Now, we should verify whether this critical point is in the interior : since $g(0,0) = 0^2+0^2-4 = -4 \leq 0$, this point is inside the feasible region.

2.2 Check the boundary

1. Define Lagrangian function:

$$\mathcal{L}(x_1, x_2, \lambda) = f(x_1, x_2) + \lambda \times g(x_1, x_2) = 4x_1^2 + 2x_2^3 + \lambda(x_1^2 + x_2^2 - 4)$$

2. Compute the partial derivatives and set them to zero:

$$\frac{\partial \mathcal{L}}{\partial x_1} = 8x_1 + 2\lambda x_1 = x_1(4 + \lambda) = 0 \tag{1}$$

Equation(1) leads to two cases:

- Case 1: $x_1 = 0$
- Case 2: $\lambda = -4$

$$\frac{\partial \mathcal{L}}{\partial x_2} = 6x_2^2 + 2\lambda x_2 = 2x_2(3x_2 + \lambda) = 0 \tag{2}$$

Equation(2) leads to two cases:

- Case 3: $x_2 = 0$
- Case 4: $\lambda = -3x_2$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = x_1^2 + x_2^2 - 4 = 0 \tag{3}$$

2.2.1 Case Analysis

Case 1: $x_1 = 0$, substitute into the constraint: $x_2^2 = 4$, we get two critical points: $(0, 2)$, $(0, -2)$.
Case 2: $\lambda = -4$, substitute into the equation(2):

$$2x_2(3x_2 - 4) = 0 \quad \Rightarrow \quad x_2 = 0 \text{ or } x_2 = \frac{4}{3}$$

Sub-case 2.1: $x_2 = 0$, substitute into equation (3) $\Rightarrow x_1^2 = 4$, so critical points are $(2, 0)$, $(-2, 0)$.
Sub-case 2.2: $x_2 = \frac{4}{3}$, substitute into equation (3):

$$x_1^2 + \left(\frac{4}{3}\right)^2 = 4 \quad \Rightarrow \quad x_1^2 = 4 - \frac{16}{9} = \frac{20}{9}$$

so we get critical points: $\left(\frac{\sqrt{20}}{3}, \frac{4}{3}\right)$, $\left(-\frac{\sqrt{20}}{3}, \frac{4}{3}\right)$.
Case 3: $x_2 = 0$. This is already covered in Case 1: $(2, 0)$, $(-2, 0)$.
Case 4: $\lambda = -3x_2$, substitute $\lambda = -3x_2$ into equation (1):

$$x_1(8 - 3x_2) = 0$$

Sub-case 4.1: $x_1 = 0$ leads to $(0, \pm 2)$, already covered.
Sub-case 4.2: $x_2 = \frac{8}{3}$, but in this case $x_2^2 > 4$, violates the constraint $x_1^2 + x_2^2 \leq 4$.

2.3 Evaluate all the critical points

Now have the following critical points:

(x_1, x_2)	$f(x_1, x_2)$
$(0, 0)$	0
$(0, 2)$	16
$(0, -2)$	-16
$(2, 0)$	16
$(-2, 0)$	16
$\left(\frac{\sqrt{20}}{3}, \frac{4}{3}\right)$	$\frac{368}{27}$
$\left(-\frac{\sqrt{20}}{3}, \frac{4}{3}\right)$	$\frac{368}{27}$

Table 2: Critical point list and value of objective function$f(x_1, x_2)$

2.4 Conclusion

For optimize $f(x_1, x_2)$ with constrain $g(x_1, x_2) = x_1^2 + x_2^2 - 4 \leq 0$

- **Maximum value**: 16 at points $(0, 2)$, $(2, 0)$, and $(-2, 0)$
- **Minimum value**: -16 at point $(0, -2)$.

3 HP1.3 Basic GA program

Note: In the experiment,

- DecodeChromosome.m: Reflect the value of gene x_1, x_2 from $[0, 1]$ to$[-5, 5]$;
- EvaluationIndividual.m: set fitness function $fitness = (g(x_1, x_2) + 1)^{-1}$;
- TournamentSelect.m: the tournament selection method used is *without replacement*, meaning that the same individual will not be chosen more than once in each tournament for comparison;
- Cross.m: for the crossover operation, a single-point crossover is employed instead of two-point crossover which is more complex;
- Mutate.m: mutation probability was not changed in the a) RunSingle.m program.

3.1 RunSingle.m

Parameter set 1 (default):

```
tournamentSize = 2;
tournamentProbability = 0.75;
crossoverProbability = 0.8;
mutationProbability = 0.02;
numberOfGenerations = 2000;
```

$g(x_1, x_2)$	Fitness	x_1	x_2
0.000000E+00	1.000000E+00	2.9999923110	0.4999981970
0.000000E+00	1.000000E+00	3.0000009537	0.5000002831
0.000000E+00	1.000000E+00	3.0000164509	0.5000044554
0.000000E+00	1.000000E+00	2.9999890327	0.4999970049
0.000000E+00	1.000000E+00	2.9999541640	0.4999886602
0.000000E+00	1.000000E+00	2.9999988675	0.4999996871
0.000000E+00	1.000000E+00	2.9999923110	0.4999981970
0.000000E+00	1.000000E+00	3.0000000596	0.4999999851
0.000000E+00	1.000000E+00	2.9999923110	0.4999981970
0.000000E+00	1.000000E+00	3.0000000596	0.4999999851

Table 3: 10 runs results from RunSingle.m(use parameter set 1)

Parameter set 2 (change tournamentSize to 4**):**

```
tournamentSize = 4;
tournamentProbability = 0.75;
crossoverProbability = 0.8;
mutationProbability = 0.02;
numberOfGenerations = 2000;
```

$g(x_1, x_2)$	Fitness	x(1)	x(2)
2.312636E-03	9.976927E-01	2.8939174978	0.4687498650
4.383197E-02	9.580086E-01	3.7500002608	0.6385599267
1.423203E-04	9.998577E-01	3.0287953624	0.5078126641
0.000000E+00	1.000000E+00	3.0004885197	0.5001212806
4.383197E-02	9.580086E-01	3.7500002608	0.6385599267
1.456212E-04	9.998544E-01	3.0273436912	0.5078126641
8.900079E-06	9.999911E-01	2.9929515121	0.4980467408
8.900079E-06	9.999911E-01	2.9929515121	0.4980467408
7.272634E-02	9.322042E-01	2.4999999255	0.3515626595
9.700094E-06	9.999903E-01	3.0078127387	0.5019311756

Table 4: 10 runs results from RunSingle.m (use parameter set 2)

3.2 RunBatch.m

Mutation Probability	Median Fitness Value
0.00	0.988401877475434
0.01	0.998244729725171
0.02	0.999999997748048
0.03	0.999999943845308
0.10	0.999972054160148
0.20	0.999614414042958
0.30	0.999310556941127
0.60	0.998660707974456
0.80	0.999039420474493
1.00	0.986571333965440

Table 5: With different mutation probability, the median fitness values result from RunBatch.m

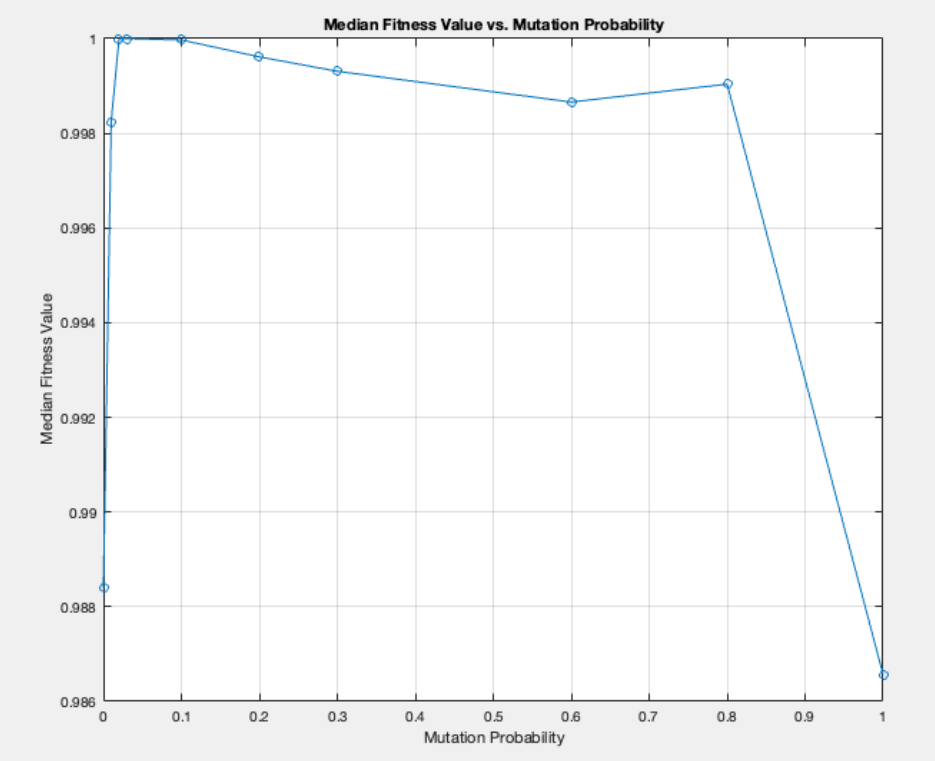

Figure 1: Median Fitness Value vs. Mutation Probability

From the Table 5 and Figure 1 we can find that

- A moderate mutation probability ($pmut = [0.02, 0.10]$) proves to be the most effective for this problem, as it balances local and global search capabilities.

- A low mutation probability ($pmut = 0.000$, which means no mutation) leads to the algorithm getting stuck in local optima.
- When $pmut > 0.10$, the fitness value begins to fluctuate and decrease, especially at $pmut = 1.000$, where the median fitness value drops to 0.990932223879751. This indicates that a high mutation rate deteriorates the solution quality of the genetic algorithm, as excessive mutation could disrupt the beneficial information of good individuals, causing a decline in fitness value.

3.3 Analytically prove the stationary point

Choosing (x_1, x_2) where $g(x_1, x_2)$ has the minimum value from Table 3 above. Since all the $g(x_1, x_2)$ values are close to zero, we simply select the values from the last row.

$$(x_1, x_2) = (3.0000000596, 0.4999999851)$$

Objective function is:

$$g(x_1, x_2) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$$

The gradient of $g(x_1, x_2)$ is

$$\frac{\partial g}{\partial x_1} = 2\times(1.5-x_1+x_1x_2)\times(-1+x_2)+2\times(2.25-x_1+x_1x_2^2)\times(-1+x_2^2)+2\times(2.625-x_1+x_1x_2^3)\times(-1+x_2^3)$$

$$\frac{\partial g}{\partial x_2} = 2\times(1.5-x_1+x_1x_2)\times x_1+2\times(2.25-x_1+x_1x_2^2)\times 2x_1x_2+2\times(2.625-x_1+x_1x_2^3)\times 3x_1x_2^2$$

At the given point $x_1 = 3.0000000596$ and $x_2 = 0.4999999851$, the partial derivatives of the objective function $g(x_1, x_2)$ are as follows:

$$\frac{\partial g}{\partial x_1} = 0.000000$$

$$\frac{\partial g}{\partial x_2} = -0.000001$$

Hence, $\nabla g = (\frac{\partial g}{\partial x_1}, \frac{\partial g}{\partial x_2})$ is close to the zero vector.
Therefore, we can conclude that $(x_1, x_2) = (3.0000000596, 0.4999999851)$ is *nearly* a stationary point of objective function $g(x_1, x_2)$.