FFR105 Stochastic optimization algorithms, 2013-09-27

Home problem 1

Simon Sigurdhsson
900322–0291

Email Sigurdhsson@gmail.com

Abstract This report discusses the solutions to home problem set 1 of the course in Stochastic optimization algorithms given by Chalmers University of Technology. In the first part, a constrained optimization problem is solved using the penalty method. In the second part, two simple optimization problems are solved using analytical methods. Finally, in the third part, a function is minimized using the standard genetic algorithm as defined by Wahde (2008, pp. 46 sqq.).

1 *Problem 1.1*

The problem is to find the minimum of the function

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

subject to the constraint

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

using the penalty method as described by Wahde (2008, pp. 30–33). The sum of the objective function in eq. (1) and the penalty term, denoted f_P, is defined as

$$f_P(\boldsymbol{x}; \mu) = \begin{cases} (x_1 - 1)^2 + 2(x_2 - 2)^2 + \mu(x_1^2 + x_2^2 - 1)^2, & x_1^2 + x_2^2 - 1 > 0, \\ (x_1 - 1)^2 + 2(x_2 - 2)^2 & \text{otherwise.} \end{cases}$$

Consequently, its gradient ∇f_P becomes

$$\nabla f_P(\boldsymbol{x}; \mu) = \begin{pmatrix} \frac{\partial f_P}{\partial x_1} \\ \frac{\partial f_P}{\partial x_2} \end{pmatrix} = \begin{pmatrix} 2(x_1 - 1) + 4\mu x_1(x_1^2 + x_2^2 - 1) \\ 4(x_2 - 2) + 4\mu x_2(x_1^2 + x_2^2 - 1) \end{pmatrix}, \tag{3}$$

which is valid for the case $x_1^2 + x_2^2 - 1 > 0$ (otherwise, simply let $\mu = 0$).

By considering the case $\mu = 0$ and setting the gradient to zero, an unconstrained minimum may be obtained. Clearly, the only stationary point of the unconstrained system is $(1, 2)$, with function value $f(1, 2) = 0$. Since other function values are larger (*e.g.* $f(1, 1) = 2$) and the function is convex on $\mathbb{R}$, the point must be a minimum.

The unconstrained minimum is a useful starting point for the penalty method. The penalty method used to solve this problem uses eq. (3) and the gradient method to obtain the minimum for several different values of the penalty parameter μ. The full implementation of the penalty method used is included with this report, with the main program residing in PenaltyMethod.m.

Table 1: *Results of optimizing eq.* (1), *constrained by eq.* (2), *using the penalty method and varying values of μ. These values were obtained with a step length $\eta = 1 \cdot 10^{-4}$ and a threshold $T = 1 \cdot 10^{-6}$.*

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
2000	0,312	0,950
2250	0,312	0,950

Table 1 lists the optima given by the penalty method for six different values of μ. The data strongly implies that the penalty method converges towards a global minimum of the constrained problem at $(x_1, x_2) \approx (0{,}312; 0{,}950)$.

2 Problem 1.2

2.1 Part A

The problem is to determine the global minimum of the function

$$f(x_1, x_2) = 4x_1^2 - x_1x_2 + 4x_2^2 - 6x_2, \tag{4}$$

subject to the constraints

$$x_1 - x_2 \leq 0, \quad x_2 - 1 \leq 0, \quad x_1, x_2 \geq 0 \tag{5}$$

using analytical methods.

First, consider the stationary points of f on the compact set defined by the constraints. It is easily found that

$$\frac{\partial f}{\partial x_1} = 8x_1 - x_2, \quad \frac{\partial f}{\partial x_2} = -x_1 + 8x_2 - 6,$$

Table 2: *Values of eq. (4) at the stationary points found.*

x_1	x_2	$f(x_1, x_2)$
$2/21$	$16/21$	$-16/7$
0	$6/8$	$-9/4$
$1/8$	1	$-33/16$
0	1	-2
$3/7$	$3/7$	$-9/7$
0	0	0
1	1	1

which yields only one stationary point, $(2/21, 16/21)$. This points clearly satisfies the constraints. Next, the stationary points on the boundaries of the constraint set are found. There points lie on the lines $(0, t)$, $(t, 1)$ and (t, t), where $0 \leq t \leq 1$. This is simple enough. In the first case, the function along the line is $f(0, t) = 4t^2 - 6t$ with the only stationary point at $t = 6/8$ corresponding to the point $(0, 6/8)$. In the second case, $f(t, 1) = 4t^2 - t - 2$ yields a stationary point at $t = 1/8$ corresponding to $(1/8, 1)$. For the last case, the function $f(t, t) = 7t^2 - 6t$ has a stationary point at $t = 3/7$, *i.e.* the point $(3/7, 3/7)$.

Table 2 shows all stationary points found, as well as the corners of the constraint set, and the corresponding values of f. Among the stationary points, f takes its lowest value at $(2/21, 16/21)$, and as such it can be concluded that the function defined by eq. (4), constrained by eq. (5), has a global minimum at $\boldsymbol{x}^* = (2/21, 16/21)$.

2.2 *Part B*

Next up is a problem concerning the function

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

constrained by the equality constrant

$$h(x_1, x_2) = x_1^2 + x_1 x_2 + x_2^2 - 21 = 0.$$

Applying the Lagrange multiplier method as described by Wahde (2008, pp. 25–28), the function

$$L(\boldsymbol{x};\lambda) = 15 + 2x_1 + 3x_2 + \lambda(x_1^2 + x_1x_2 + x_2^2 - 21) \tag{6}$$

is obtained, and the gradient of eq. (6) is easily computed as

$$\frac{\partial L}{\partial x_1} = 2 + \lambda(2x_1 + x_2), \tag{7a}$$

$$\frac{\partial L}{\partial x_1} = 3 + \lambda(x_1 + 2x_2), \tag{7b}$$

$$\frac{\partial L}{\partial x_1} = x_1^2 + x_1x_2 + x_2^2 - 21. \tag{7c}$$

Setting the gradient to zero and manipulating eqs. (7a) and (7b), expressions for the two original coordinates are obtained as $x_1 = -\tfrac{4}{3}\lambda^{-1}$ and $x_2 = -\tfrac{1}{3}\lambda^{-1}$. These expressions may be inserted into eq. (7c), which then yields $\lambda = \pm\tfrac{1}{3}$.

The two stationary points obtained using the Lagrange method, $(1, 4)$ and $(-1, -4)$, have function values 29 and 1 respectively. As such, it is clear that the constrained function f has a minimum at $\boldsymbol{x}^* = (-1, -4)$.

3 Problem 1.3

3.1 Part A

The first part of the final problem is to implement the standard genetic algorithm as described by Wahde (2008, p. 56). The implementation includes tournament

Table 3: *Parameter values used in the genetic algorithm. The number of generations is set so that the number of evaluations remains fixed at* 10 000.

m	N	p_{tour}	p_c	p_{mut}	j
50	100	0,75	0,80	0,02	2

Table 4: *Results of running the genetic algorithm with the parameters given in table 3. Data from 100 optimization rounds.*

Best value	Best point	Worst value	Mean	Median
3,000	(0,000; −1,000)	85,094	3,822	3,000

selection with j rounds and a parameter p_{tour}, mutation with a probability p_{mut} and a crossover probability p_c. In addition, the implementation also includes elitism, with a single copy of the best individual being included in the subsequent generation. Table 3 lists the parameters used in the implementation, with m and N denoting the chromosome length and population size, respectively. As noted in table 3, the number of generations depends on the population size in order to keep the number of evaluations constant.

The complete implementation is included with this report, the main program file being FunctionOptimization.m.

3.2 *Part B*

The objective function used to test the implementation is

$$\begin{aligned} g(x_1, x_2) = {} & \left(1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right) \times \\ & \left(30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\right) . \end{aligned} \quad (8)$$

Table 4 lists the results of running the algorithm with the objective function in eq. (8) 100 times. The best minimum found by the algorithm is located at $\boldsymbol{x}^* = (0{,}000; -1{,}000)$ and has function value 3,000. Since the median of the minima matches this value, an obvious conclusion is that this optimum is found more than half the time, and that it therefore should be a good minimum. This will be discussed further in section 3.3.

As is evident by the worst value found by the algorithm, not all rounds result in finding this global optimum. A back-of-the-envelope calculation using the mean indicates that the algorithm has suffered from premature convergence once or twice, *i.e.* 1–2 % of the time.

3.2.1 *Parameter evaluation*

Since the algorithm has a relatively high failure rate, an analysis of the parameter values is in order. Tables 5 to 9 show the result of optimizing the objective function from eq. (8), while varying some of the parameters from table 3 (one at a time). As with the data in table 4, the data is the result of running the algorithm 100 times for each parameter set.

Simple statistical analysis has been applied to the data, testing the null hypothesis H_0 that the distribution of data with a varied parameter is equal to the distribution of the data in table 4 (*i.e.* from standard parameters). The hypothesis test carried out is the two-sample Kolmogorov-Smirnov test (described by Massey (1951) and implemented in MATLAB as KSTEST2, *cf.* The Mathworks, Inc. (2013)), with 95 % confidence.

The parameter variations were performed under the assumption that the parameters are relatively robust (*i.e.* insensitive to small changes in value), and therefore only large variations were included. For some parameters, *e.g.* mutation probability and tournament selection parameter, this yields unreasonable results, indicating that the parameter isn't very robust on a larger scale.

Table 5: *Results of running the genetic algorithm with varying tournament size. Data from 100 optimization rounds.*

j	Best value	Best point	Worst value	Mean	Median	H_0 rejected
0	3,000	(0,001; −1,000)	5,586	3,189	3,029	Yes
1	3,000	(0,000; −1,000)	3,002	3,000	3,000	Yes
2	3,000	(0,000; −1,000)	4,699	3,018	3,000	No
5	3,000	(0,000; −1,000)	89,822	7,306	3,000	Yes
10	3,000	(0,000; −1,000)	54,900	3,604	3,000	Yes

Tournament size j Table 5 contains data obtained by varying the tournament size. It is apparent that a tournament size of 0 (selecting individuals at random, disregarding fitness) results in poor perfomance, as indicated by the rejected hypothesis test. Large tournaments ($j > 2$) perform fairly well, but are still worse than the case $j = 2$, and the null hypothesis is rejected.

The tournament size $j = 1$ constitutes an interesting case. Although the null hypothesis is in fact rejected, it is evident that the actual results are better than the results in table 4. The reason the null hypothesis is rejected is that the alternative, H_1, is *two-sided* (*i.e.* H_1 represents the alternative hypothesis "the tail of the distribution is larger *or* smaller"), and in this case the tail of the distribution is *smaller* (hence, better).

Tournament selection parameter p_{tour} The effects of varying the tournament selection parameter are shown by table 6. The only case for which the null hypothesis is not rejected is $p_{\text{tour}} = 0.75$, the original parameter value. This indicates that the algorithm, when used on this particular problem, is sensitive to the value of this parameter. The median and mean of the cases for which the null hypothesis is rejected indicate that these parameter values are in fact worse than the original one.

In particular, the case $p_{\text{tour}} = 0$ (always selecting the weaker individual) has very poor performance — in fact, the real minimum is never found in this case. Similarly, the case $p_{\text{tour}} = 1/4$ has poor performance. Of course, preferring the weaker individual makes little sense and this is why these parameter values have such poor performance.

Table 6: *Results of running the genetic algorithm with varying tournament selection parameter. Data from 100 optimization rounds.*

p_{tour}	Best value	Best point	Worst value	Mean	Median	H_0 rejected
0,00	4,468	$(-0{,}060; -0{,}973)$	266,330	54,964	41,803	Yes
0,25	3,003	$(-0{,}000; -1{,}003)$	97,603	23,880	15,158	Yes
0,50	3,000	$(0{,}000; -1{,}000)$	4,956	3,108	3,023	Yes
0,75	3,000	$(0{,}000; -1{,}000)$	3,018	3,000	3,000	No
1,00	3,000	$(0{,}000; -1{,}000)$	84,009	5,613	3,000	Yes

Table 7: *Results of running the genetic algorithm with varying mutation probability. Data from 100 optimization rounds.*

p_{mut}	Best value	Best point	Worst value	Mean	Median	H_0 rejected
0,00	3,000	(0,000; −1,000)	98,994	14,190	4,805	Yes
0,02	3,000	(0,000; −1,000)	84,578	3,851	3,000	No
0,50	3,003	(0,003; −1,001)	8,946	4,046	3,704	Yes
0,75	3,003	(0,002; −0,997)	7,582	3,969	3,562	Yes
1,00	3,007	(0,006; −0,999)	31,607	8,371	4,749	Yes

Surprisingly, the case $p_{\text{tour}} = 1$ (never picking the weaker individual) also rejects the null hypothesis (and has higher median and mean). This is likely due to premature convergence. The algorithm still performs fairly well in this case, as indicated by the median being close to the best value found.

Mutation probability p_{mut} Mutation has the important effect of introducing new genetic material into the population, thereby somewhat limiting the risk of premature convergence. A large mutation probability may have the adverse effect of destroying "good" genetic material. As shown by table 7, the algorithm is very sensitive to large changes in mutation probability. The only case for which the null hypothesis isn't rejected is the original parameter value, $p_{\text{mut}} = m^{-1}$. This is in accordance with theoretical results.

The only other cases with even remotely acceptable performance are $p_{\text{mut}} = 1/2$ and $p_{\text{mut}} = 3/4$. For $p_{\text{mut}} = 0$ the results indicate that the algorithm suffers from premature convergence a lot of the time, which would be expected since no "new" genetic material is introduced.

Crossover probability p_c As indicated by table 8, the algorithm is fairly insensitive to changes in the crossover probability. The null hypothesis is only rejected for $p_c = 1/5$, and the median is equal to the best value in all cases. The rejected case will most likely result in genetic material spreading very slowly, which means more generations are required to reach the global optimum.

Since elitism has been applied, there is no risk of completely losing the best individual, which may be why the case $p_c = 1$ doesn't have poor performance.

Table 8: *Results of running the genetic algorithm with varying crossover probability. Data from 100 optimization rounds.*

p_c	Best value	Best point	Worst value	Mean	Median	H_0 rejected
0,20	3,000	(0,000; −1,000)	84,001	4,162	3,000	Yes
0,40	3,000	(0,000; −1,000)	33,910	3,646	3,000	No
0,60	3,000	(0,000; −1,000)	3,174	3,002	3,000	No
0,80	3,000	(0,000; −1,000)	4,581	3,017	3,000	No
1,00	3,000	(0,000; −1,000)	84,000	3,810	3,000	No

With no elitism, there is great potential to destroy "good" genetic material with a high p_c. Conversely, a low p_c will result in slower convergence, since the algorithm will be driven mainly by mutation.

One may also observe that the case $p_c = 3/5$ looks like it has slightly better performance than the original parameter value $p_c = 4/5$. The difference isn't significant (in that case, the null hypothesis would be rejected as discussed earlier), but it may be worth investigating further.

Population size N Table 9 shows data obtained by varying the population size. In order to keep the results comparable, the number of evaluations has been fixed at 10 000, which means the number of generations g varies with population size.

Again, the only case for which the null hypothesis is not rejected is the original parameter value. Looking at the mean and median, the case $N = 50$ has acceptable performance as well, while $N = 10$ often doesn't converge (*viz.* the worst value found is very large, as is the mean).

Larger populations have slightly better performance, which is likely due to them "blanketing" the region with individuals, covering much of the domain. This means the initial population likely has individuals close to the optimum, while they don't have enough generations to converge all the way.

Table 9: *Results of running the genetic algorithm with varying population size and number of generations, such that the number of evaluations remain fixed at* 10 000. *Data from 100 optimization rounds.*

N	Best value	Best point	Worst value	Mean	Median	H_0 rejected
10	3,000	$(0{,}000; -1{,}000)$	249,594	50,004	4,581	Yes
50	3,000	$(0{,}000; -1{,}000)$	84,000	3,942	3,000	Yes
100	3,000	$(0{,}000; -1{,}000)$	4,012	3,012	3,000	No
500	3,000	$(-0{,}000; -1{,}000)$	3,625	3,034	3,011	Yes
1000	3,001	$(0{,}001; -0{,}999)$	4,459	3,200	3,123	Yes

3.3 Part C

The final issue at hand is to prove, *analytically*, that the function given by eq. (8) has a stationary point at $\boldsymbol{x}^* = (0{,}000; -1{,}000)$, the point obtained through stochastic optimization using the standard genetic algorithm.

Consider $g(\boldsymbol{x}) = (1 + g_1 g_2)(30 + g_3 g_4)$ with

$$\begin{aligned} g_1(\boldsymbol{x}) &= (1 + x_1 + x_2)^2, \quad g_3(\boldsymbol{x}) = (2x_1 - 3x_2)^2, \\ g_2(\boldsymbol{x}) &= (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2), \\ g_4(\boldsymbol{x}) &= (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2), \end{aligned}$$

which is a rephrasing of eq. (8) in terms of four subsidiary functions. It is trivially known that the derivative of g may be written $g' = (1 + g_1g_2)(g_3'g_4 + g_3g_4') + (g_1'g_2 + g_1g_2')(30 + g_3g_4)$, which may be extended to the gradient as

$$\nabla g(\boldsymbol{x}) = \begin{pmatrix} (1 + g_1g_2)\left(\dfrac{\partial g_3}{\partial x_1}g_4 + g_3\dfrac{\partial g_4}{\partial x_1}\right) + \left(\dfrac{\partial g_1}{\partial x_1}g_2 + g_1\dfrac{\partial g_2}{\partial x_1}\right)(30 + g_3g_4) \\ (1 + g_1g_2)\left(\dfrac{\partial g_3}{\partial x_2}g_4 + g_3\dfrac{\partial g_4}{\partial x_2}\right) + \left(\dfrac{\partial g_1}{\partial x_2}g_2 + g_1\dfrac{\partial g_2}{\partial x_2}\right)(30 + g_3g_4) \end{pmatrix}.$$

Partial derivatives of the subsidiay functions may be computed as

$$\begin{aligned}
\frac{\partial g_1}{\partial x_1} &= \frac{\partial g_1}{\partial x_2} = 2x_1 + 2x_2 + 2, & \frac{\partial g_2}{\partial x_1} &= \frac{\partial g_2}{\partial x_2} = 6x_1 + 6x_2 - 14, \\
\frac{\partial g_3}{\partial x_1} &= 8x_1 - 12x_2, & \frac{\partial g_3}{\partial x_2} &= -12x_1 + 18x_2, \\
\frac{\partial g_4}{\partial x_1} &= 24x_1 - 36x_2 - 32, & \frac{\partial g_4}{\partial x_2} &= -36x_1 + 54x_2 + 48.
\end{aligned}$$

To avoid the considerable headache of inserting these equations directly into the rewritten g (since the objective is to assert that $\boldsymbol{x}^*$ is a stationary point, not to provide a full analytic expression for ∇g), numerical values for these functions at $\boldsymbol{x}^*$ are obtained as

$$\begin{aligned}
g_1(\boldsymbol{x}^*) &= 0, & g_2(\boldsymbol{x}^*) &= 36, & g_3(\boldsymbol{x}^*) &= 9, & g_4(\boldsymbol{x}^*) &= -3, \\
\frac{\partial g_1}{\partial x_1}(\boldsymbol{x}^*) &= 0, & \frac{\partial g_2}{\partial x_1}(\boldsymbol{x}^*) &= -8, & \frac{\partial g_3}{\partial x_1}(\boldsymbol{x}^*) &= 12, & \frac{\partial g_4}{\partial x_1}(\boldsymbol{x}^*) &= 4, \\
\frac{\partial g_1}{\partial x_2}(\boldsymbol{x}^*) &= 0, & \frac{\partial g_2}{\partial x_2}(\boldsymbol{x}^*) &= -8, & \frac{\partial g_3}{\partial x_2}(\boldsymbol{x}^*) &= -18, & \frac{\partial g_4}{\partial x_2}(\boldsymbol{x}^*) &= -6.
\end{aligned}$$

This yields the gradient

$$\nabla g(\boldsymbol{x}) = \begin{pmatrix} 1 \cdot (-12 \cdot 3 + 9 \cdot 4) + 0 \cdot (30 - 9 \cdot 3) \\ 1 \cdot (18 \cdot 3 - 9 \cdot 6) + 0 \cdot (30 - 9 \cdot 3) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

which shows that $\boldsymbol{x}^*$ is a stationary point as desired, with $g(\boldsymbol{x}^*) = 3$.

References

Massey, Frank J., Jr. 1951. “The Kolmogorov-Smirnov Test for Goodness of Fit”. *Journal of the American Statistical Association* 46 (253): 68–78.

The Mathworks, Inc. 2013. *Two-sample Kolmogorov-Smirnov test.* Accessed 24th September 2013. http://www.mathworks.se/help/stats/kstest2.html.

Wahde, M. 2008. *Biologically Inspired Optimization Methods: An Introduction.* 1st ed. Southampton, Boston: WIT Press. ISBN: 978-1-84564-148-1.