16.888/ESD 77 Multidisciplinary System Design Optimization:
Assignment 4 Part a) Solution

Part A1

(a) Increasing mutation rate results in increased population density.

(b) The required number of bits:

$$nbits = f\left(n, \Delta x, d\right) = \sum_{i=1}^{n} \left\lceil \ln\left[\left(\frac{x_{UB} - x_{LB}}{\Delta x}\right)\right] \Big/ \ln d \right\rceil = 17$$

(c) Using the same formula as above, we get: nbits = 2.

(d) The binary to decimal conversion can be written as:

$$d = (\text{binary to decimal conversion}) + x_{\min}$$

Here $x_{\min}$ =1 and therefore to get a decimal number 3, we should look for a binary string that represent decimal number 2. The corresponding binary string is 10 (i.e., $1.2^{2-1} + 0.2^{0} = 2$).

(e) The correct answer is (b) – B&C.
A is not possible because it's starting bit is 1, but both parents starting bit is 0. D is not possible since it's 4th bit is 1, but that of both parents is 0.

(f) $p_{F_1} = \dfrac{F_1}{\sum_{i=1}^{5} F_i} = \dfrac{100}{1000} = 0.10$.

(g) In this population of size 5, only F_2 has a better fitness than F_1. Hence, assuming week dominance, F_2 would be selected in 4 out of 5 instances. The only time it would lose when it is compared to F_2. Therefore the probability that F_1 would be selected is 4/5 = 0.80.

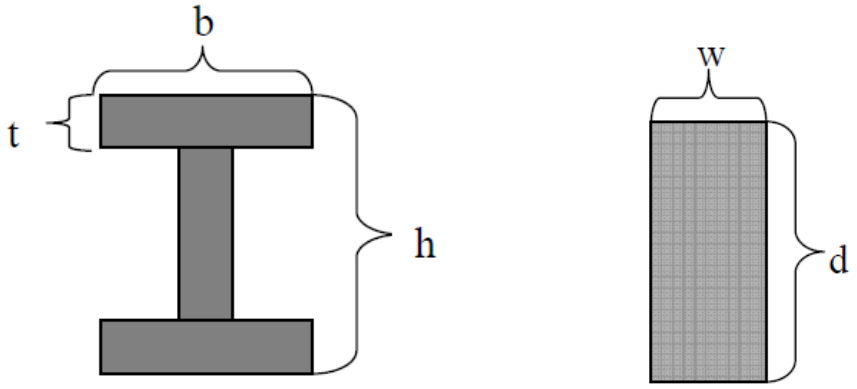
(h) A 16 bit variable can have 2^{16} possible values and a 4 bit variable can have 2^4 possible values.
Hence the total number of possible population members = 2^{16}. 2^4=2^{20}.

Part A2

This problem has 5 continuous variables (i.e., the c/s dimensions of I beams and supporting column) and 3 integer variables (i.e., number of beams and material identifiers). Therefore, we can use heuristic algorithms like Simulated Annealing, Genetic Algorithm; or Branch and Bound Algorithms, or perform explicit enumeration on integer variables while optimizing for 5 continuous variables (this would mean 4*4*4 = 64 gradient-based optimization runs for each possible inter parameter settings).

This problem also brings out the role of constraints and bounding characteristics in MDO problem formulations.

The first step would be to use some "physical" bounds on the cross-sectional dimensions. For example, one cannot produce an I-beam of thickness 0 or 1 mm or it is impractical for the height of I-beam to go over say, 5 m in an over-bridge construction.



In addition, for it to be an I-beam, impose two geometrical constraints:

$$\frac{2t}{h} \leq 1$$

$$\frac{t}{b} \leq 1$$

If there is no positive lower limit on the beam height, it will lead to non-physical/degenerate solution since bending stress is undefined (i.e., would be zero) if the height of the beam is zero. In that case, imposing a constraint on shear stress will ensure that you never have a "virtual" beam. The optimal bridge might vary depending on the bounds selected.

Let us define the optimization problem assuming the bounds as detailed below:

$$
\begin{aligned}
&\textit{Min.}\ \text{Cost, C} = f(X) \\
&s.t\ \ \frac{2t}{h} \le 1 \\
&\quad \frac{t}{b} \le 1 \\
&\quad \sigma_{bending} \le \sigma_{failure,bending} \\
&\quad \sigma_{shear} \le \sigma_{failure,shear} \\
&\quad \sigma_{\sup port} \le \sigma_{failure,\sup port} \\
&\quad P_{applied} \le P_{critical,\sup port} \\
&\quad 0.1\ \text{m} \le b \le 1\ \text{m} \\
&\quad 0.01\ \text{m} \le t \le 0.5\ \text{m} \\
&\quad 0.1\ \text{m} \le h \le 2\ \text{m} \\
&\quad 0.2\ \text{m} \le w \le 2\ \text{m} \\
&\quad 0.3\ \text{m} \le d \le 3\ \text{m}
\end{aligned}
$$

The bounds can be chosen to be different than the above but should be physical realizable.

One might require that there be no beam overhang (which lead to structural problems not considered in this problem) and in that the depth of the support (d) should be higher that the combined flange width (b) of beams. This can be added as another constraint to the above list of the form (# beams)*b+ε)≤d and in the example shown later, ε = 0.05 m was used. This ensures that there is that the beams completely rest on the supporting column.

(a) This problem consists of continuous and integer design variables and we cannot directly apply gradient-based optimization techniques that handle continuous variables only. Therefore, we can use heuristic algorithms like Simulated Annealing, Genetic Algorithm; or Branch and Bound Algorithms, or perform explicit enumeration on integer variables while optimizing for 5 continuous variables (this would mean 4*4*4 = 64 gradient-based optimization runs for each possible inter parameter settings).

The solutions based on GA and explicit enumeration on integer variables is presented below.

(b) If we use explicit enumeration of three integer variables, we have to run 4*4*4 = 64 gradient-based optimization using 5 continuous variables. The result of such an analysis is shown below:

Problem formulation with SQP	b (m)	t (m)	h (m)	w (m)	d (m)	Beam material	Support Material	# beams	Optimal Cost
w/o shear stress	0.3471	0.01	2.0	0.374	0.374	A514	Concrete	1	$5771.3

constraint									
With shear stress constraint	0.3471	0.01	2.0	0.374	0.374	A514	Concrete	1	$5771.3
With constraint linking beam width (b), number of beams (N) and support depth (d)	0.3471	0.01	2.0	0.3664	0.3971	A514	Concrete	1	$5774.1

One can observe that the shear constraint is not active for the set of variable bounds used and therefore do not influence the optimal solution. The result would be different (and shear constraint would be active) if beam height lower limit is zero. When the constraint linking the overall beam flange length and support depth is used, this constraint is active at the optimal solution. Notice that the cost has increased from $5771.3 to $5774.1.

Now using GA with higher initial penalty, lower crossover fraction of 0.6 with increases mutation rate (generations = 50, popsize = 200), the following solution was obtained:

Problem formulation with GA	b (m)	t (m)	h (m)	w (m)	d (m)	Beam material	Support Material	# beams	Optimal Cost
With shear stress constraint	0.3744	0.01	1.938	0.374	0.374	A514	Concrete	1	$5796.6
With constraint linking beam width (b), # beams (N) and support depth (d)	0.4153	0.01	1.865	0.3476	0.4653	A514	Concrete	1	$5813.4

The GA closed on the near optimal solution within first 20 generations but had hard time finding the final solution. The solutions obtained by GA are within 1% to that obtained using explicit enumeration on integer variables.

It is worth noting that a hybrid strategy of running GA for 20 generations and handing over the GA solution (with integer variables held fixed and using 5 continuous variables as the design variables) as the starting point for gradient-based algorithm obtained the optimal solution much faster.

Part A3

The gradient of the objective function is:

$$\nabla f = [0.0001x_1 \;\; 0.001x_2 \;\; 0.01x_3 \;\; 0.1x_4 \;\; x_5 \;\; 10x_6 \;\; 100x_7 \;\; 1000x_8]^T$$

The Hessian matrix is diagonal and constant:

$$H = \begin{bmatrix} 0.0001 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.001 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1000 \end{bmatrix}$$

In order to compare the algorithms appropriately, a set of 10 random realizations in the design space is generated and used as the initial points in all cases.

Start points	X(1)	X(2)	X(3)	X(4)	X(5)	X(6)	X(7)	X(8)
1	-0.7824	4.1574	2.9221	4.5949	1.5574	-4.6429	3.4913	4.3399
2	1.7874	2.5774	2.4313	-1.0777	1.5548	-3.2881	2.0605	-4.6817
3	-2.2308	-4.5383	-4.0287	3.2346	1.9483	-1.8290	4.5022	-4.6555
4	-0.6126	-1.1844	2.6552	2.9520	-3.1313	-0.1024	-0.5441	1.4631
5	2.0936	2.5469	-2.2397	1.7970	1.5510	-3.3739	-3.8100	-0.0164
6	4.5974	-1.5961	0.8527	-2.7619	2.5127	-2.4490	0.0596	1.9908
7	3.9090	4.5929	0.4722	-3.6138	-3.5071	-2.4249	3.4072	-2.4572
8	3.1428	-2.5648	4.2926	-1.5002	-3.0340	-2.4892	1.1604	-0.2671
9	-1.4834	3.3083	0.8526	0.4972	4.1719	-2.1416	2.5720	2.5373
10	-1.1955	0.6782	-4.2415	-4.4605	0.3080	2.7917	4.3401	-3.7009

(a) In all 10 cases, Newton method converged in 1 step. This is understandable because the objective function is quadratic with constant Hessian matrix and Newton method is guaranteed to converge in one step in such cases.

(b) Conjugate gradient method: Exact line search: 14 iterations to converge on average; Inexact line search using fmincon: 34 iterations to converge on average.

(c) Steepest descent method: Did not converge in any of the cases even after 5000 iterations.

(d) Condition number of Hessian, $\kappa(H) = \dfrac{\lambda_{\max}}{\lambda_{\min}} = \dfrac{1000}{10^{-4}} = 10^7$.

(e)
Non-singular transformation matrix for hessian conditioning/scaling of the form x = Ly:

We can write, objective function $f(x) = \dfrac{1}{2} x^T H x = \dfrac{1}{2} y^T \underbrace{L^T H L}_{=I} y \equiv \dfrac{1}{2} y^T y$.

So the transformation matrix L must satisfy the relationship, $L^T HL = I$ (identity matrix with all eigenvalues equal to 1). Utilizing the fact that H is diagonal and the diagonal entries represent its eigenvalues, we can compute the transformation matrix as:

$$L = \begin{bmatrix} \frac{1}{\sqrt{0.0001}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{0.001}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{0.01}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\sqrt{0.1}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{10}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{100}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{1000}} \end{bmatrix}$$

(f) Number of iteration to converge before and after scaling:

Algorithm	$N_{original}$	N_{scaled}	Improvement (%)
Newton	1	1	No impact
Conjugate gradient/Quasi-Newton (fmincon)	14/36	2	85(%)/ 95(%)
Steepest Descent	>5000	2	~100(%)

(g) Comments on sensitivity of these algorithms to scaling: Newton method is insensitive to scaling while steepest descent is highly sensitive. Since the objective function is quadratic, Newton methods converges in just one step in any case. The conjugate gradient and the quasi-Newton method (with identity matrix as the initial Hessian and inexact line search) converged in two steps on scaling. The steepest descent method becomes a viable option only after scaling, which makes the Hessian matrix an identity. Therefore the scaling operation is mandatory for using steepest descent method in this case.

ESD.77 / 16.888 Multidisciplinary System Design Optimization
Spring 2010