ME697Y Intelligent Systems – Project2

Extended Evolutionary Strategy

Bin Zhang

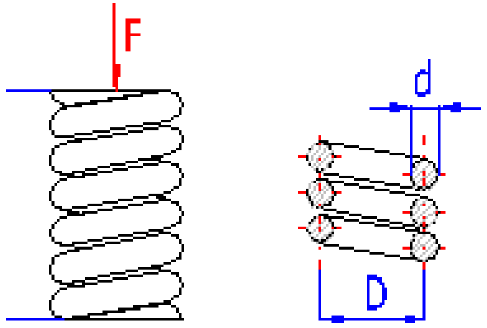
Problem Statement

- Design of a coil spring system with **least material** (minimum volume)

$$\text{minimize } f(\mathbf{x}, \mathbf{d}) = \pi\, x_1 d_2^2 (d_1 + 2)/4$$

- **Design variables:**
  - ***x1***: ***D***, the winding diameter, continuous variable
  - ***d1***: ***N***, the number of spring coils, integer, *1,2,3,.....*
  - ***d2***: ***d***, the wire diameter, which can only take the following 42 values:

0.009	.0095	.0104	.0118	.0128	.0132
.014	.015	.0162	.0173	.018	.020
.023	.025	.028	.032	.035	.041
.047	.054	.063	.072	.080	.092
.105	.120	.135	.148	.162	.177
.192	.207	.225	.244	.263	.283
.307	.331	.362	.394	.4375	.500



Problem Statement

- **Constraints:** maximum shear strength, maximum free length, maximum deflection under preload, deflection from preload to maximum load, etc.

$$g_1(\mathbf{x},\mathbf{d}) = S - \frac{8C_f F_{max} x_1}{\pi d_2^3} \geq 0$$

$$g_2(\mathbf{x},\mathbf{d}) = l_{max} - l_f \geq 0$$

$$g_3(\mathbf{x},\mathbf{d}) = d_2 - d_{min} \geq 0$$

$$g_4(\mathbf{x},\mathbf{d}) = D_{max} - x_1 - d_2 \geq 0$$

$$g_5(\mathbf{x},\mathbf{d}) = \frac{x_1}{d_2} - 3.0 \geq 0$$

$$g_6(\mathbf{x},\mathbf{d}) = \delta_{pm} - \delta_p \geq 0$$

$$g_7(\mathbf{x},\mathbf{d}) = l_f - \delta_p - \frac{F_{max} - F_p}{K} - 1.05(d_1 + 2)d_2 \geq 0$$

$$g_8(\mathbf{x},\mathbf{d}) = -\delta_w - \frac{F_{max} - F_p}{K} \geq 0$$

$$C_f = \frac{4(x_1/d_2) - 1}{4(x_1/d_2) - 4} - \frac{0.165 d_2}{x_1}$$

$$K = \frac{G d_2^4}{8 d_1 x_1^3}$$

$$l_f = \frac{F_{max}}{K} + 1.05(d_1 + 2)d_2$$

$$\delta_p = \frac{F_p}{K}$$

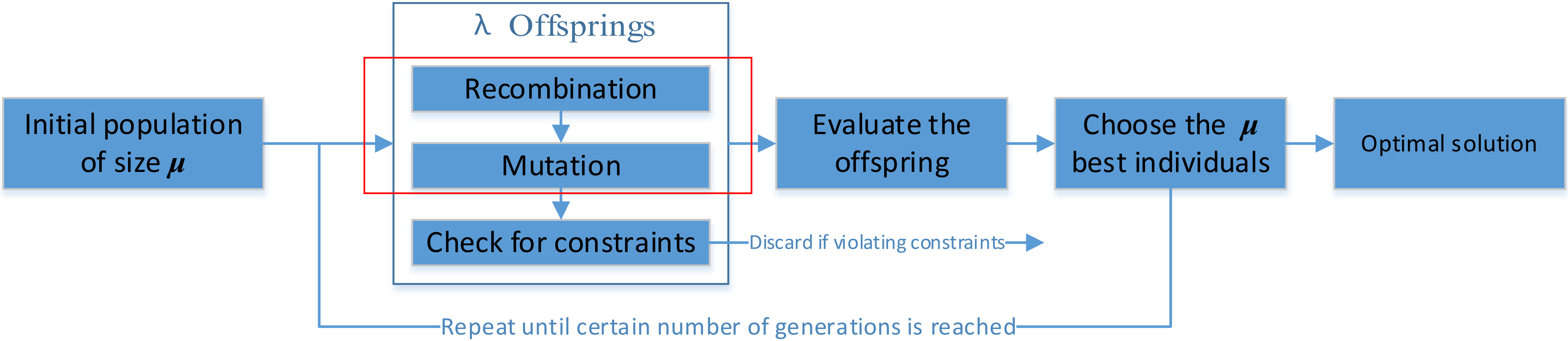
$S = 189000\,psi$ $\quad F_{max} = 1000lb$ $\quad l_{max} = 14in.$

$d_{min} = 0.2in.$ $\quad D_{max} = 3.0in.$ $\quad F_p = 300lb$

$\delta_{pm} = 6.0in.$ $\quad \delta_w = 1.25in.$ $\quad G = 11.5\times10^6\,psi$

Extended Evolutionary Strategy

- The original evolutionary strategy was developed for optimization problems with continuous variables.
- How could we handle continuous variables (***x1***), integer variables (***d1***) and discrete variables (***d2***) simultaneously?
- **Extended (*μ*, *λ*) ES:**



Extended Evolutionary Strategy

- Each individual consists not only of the design variables ***x*** and ***d***, but also of the control vectors ξ and $\mathbf{s}$, which controls the step size of mutation.

$$\mathbf{a} = \{(\mathbf{x}, \boldsymbol{\xi}), (\mathbf{d}, \mathbf{s})\}$$

- **Continuous Variables:**

Recombination	Mutation
Parents: $(\mathbf{x}_S, \boldsymbol{\xi}_S)$ and $(\mathbf{x}_T, \boldsymbol{\xi}_T)$ Offspring: $\begin{cases} x_i = x_{S,i} \text{ or } x_{T,i} \\ \xi_i = \dfrac{\xi_{S,i} + \xi_{T,i}}{2} \end{cases}$	$\xi_i' = \xi_i \cdot \exp\left(\tau' \cdot N(0,1) + \tau \cdot N(0,1)\right)$ $x_i' = x_i + \xi_i' \cdot N(0,1)$ $\tau \propto \dfrac{1}{\sqrt{2\sqrt{n_x}}}, \quad \tau' \propto \dfrac{1}{\sqrt{2n_x}}$

N(0,1) is a normally distributed random variable with expectation 0 and standard deviation 1

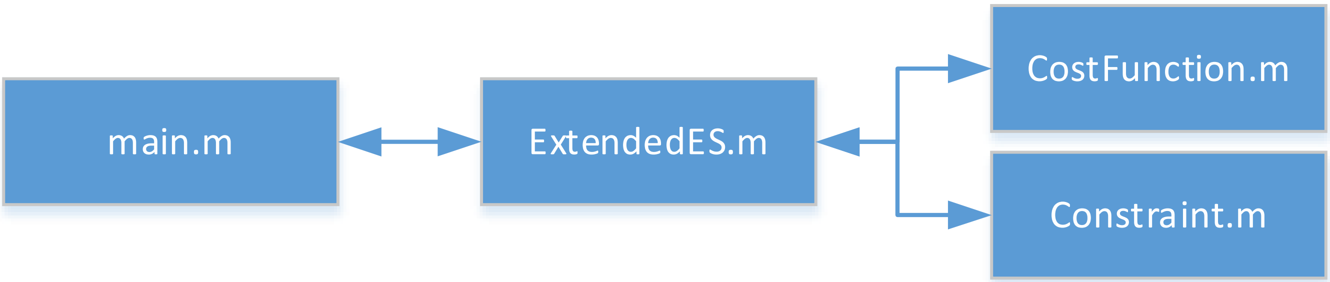
Extended Evolutionary Strategy

- **Discrete Variables and Integer Variables:**
  - Map the discrete values to integer indices *{1,2,…nj}*
  - The integer variable ***d1*** is mapped to itself
  - The mutation operation is based on the integer index

$$\begin{matrix} 0.009 & 0.0095 & \cdots & 0.500 \\ & \Updownarrow & & \\ 1 & 2 & \cdots & 42 \end{matrix}$$

Recombination	Mutation
Parents: $(\mathbf{d}_S, \mathbf{s}_S)$ and $(\mathbf{d}_T, \mathbf{s}_T)$ Offspring: $\begin{cases} d_i = d_{S,i} \text{ or } d_{T,i} \\ s_i = \dfrac{s_{S,i} + s_{T,i}}{2} \end{cases}$	$s' = s \cdot \exp(\tau'' \cdot N(0,1)) \quad \tau'' = \dfrac{1}{\sqrt{n_d}}$ $p = 1 - \dfrac{s/n_d}{(1 + (s/n_d)^2)^{1/2} + 1}$ $P\{z_j = k\} = \dfrac{p}{2-p}(1-p)^{\vert k \vert}, \quad k \in \mathrm{Z},$ $d_j' = d_j + z_j, \quad j = 1, \ldots, n_d,$ *P{zj=k}* is the possibility for *zj* to take the value of k.

MATLAB Program



- **function [best,mean_cost,min_cost] = ExtendedES(lambda, miu, generations,xlim,dlim,D)**

  % lambda: offspring population

  % miu: parent population

  % generations: number of generations

  % xlim: a nx by 2 matrix containing the limit of continuous variables

  % dlim: a nd by 2 matrix containing the limit of discrete variables

  % D: ceil array of the indexed values of the discrete variables

- For the coil compression spring design problem

$$\mu = 150 \quad \lambda = 1000 \quad generations = 20$$

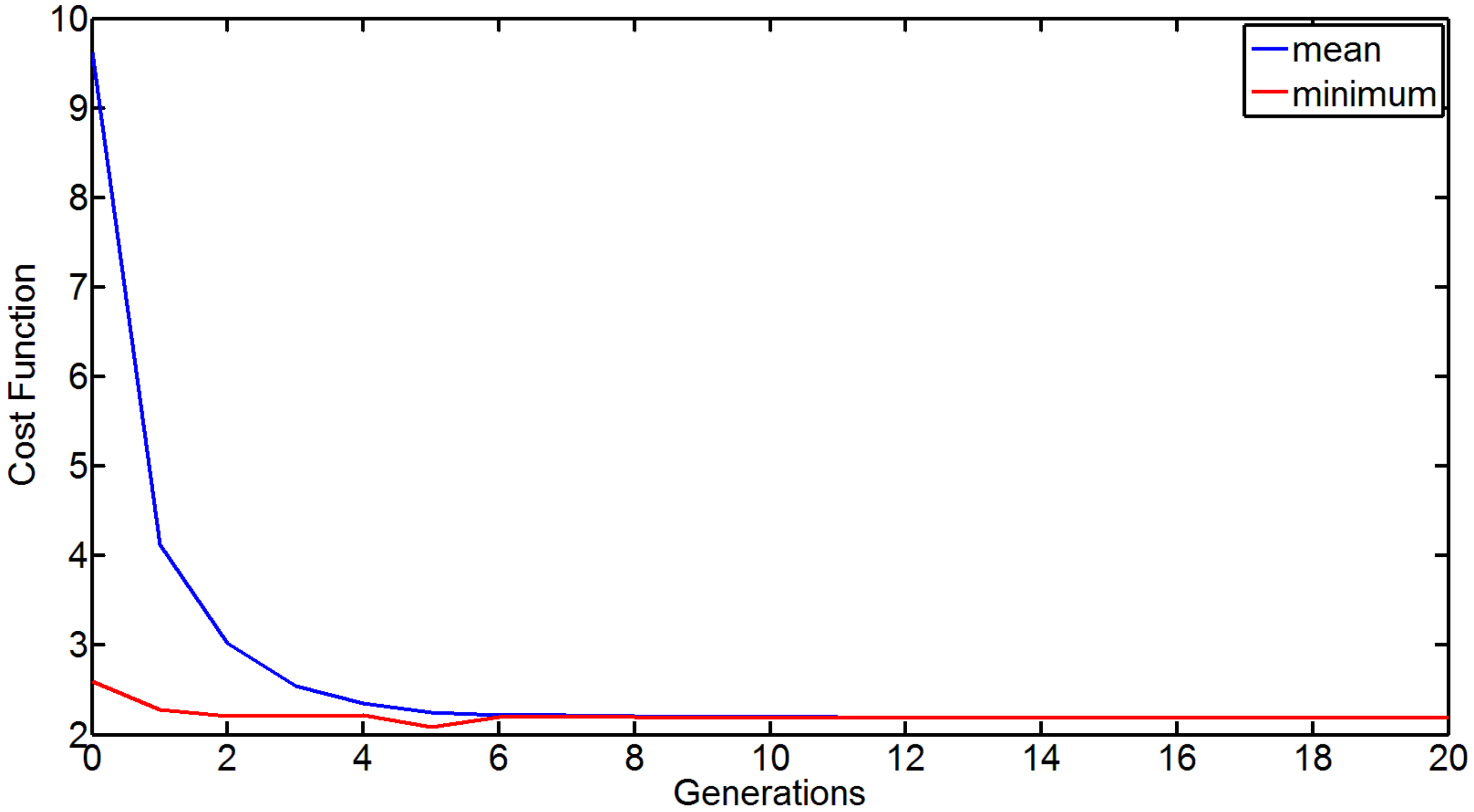
$$0 < x_1 < 5 \quad 1 < d_1 < 25$$

Optimization Results

Optimum Solution of the Spring Design

Items	Sandgren (1990)	Chen and Tsao (1993)	Meta-GA (Wu and Chow, 1995)	ES	My results
x_1 (in.)	1.180701	1.2287	1.227411	1.10918	1.109175
d_1 (number of spring coils)	10	9	9	9	9
d_2 (in.)	0.283	0.283	0.283	0.263	0.263
g_1 (psi)	54.309	415.969	550.993	3615.633	3616.310
g_2 (in.)	8.8187	8.9207	8.9264	9.176612	9.176636
g_3 (in.)	0.08298	0.08300	0.08300	0.063000	0.063000
g_4 (in.)	1.8193	1.7713	1.7726	1.627820	1.627825
g_5	1.1723	1.3417	1.3371	1.217414	1.217396
g_6 (in.)	5.4643	5.4568	5.4585	5.464279	5.464286
g_7 (in.)	0.0	0.0	0.0	0.0	0.0
g_8 (in.)	0.0004	0.0174	0.0134	0.000017	6.039e-10
$f(\mathbf{x}, \mathbf{d})$ (in.^{3})	2.7995	2.6709	2.6681	2.0823	2.0823

Optimization Results



Discussion

- By enumerative searching with ***x1*** resolution of ***0.0001***, the solution *{Spring diameter: 1.1092; Number of springs: 9; Wire diameter: 0.2630; Min f = 2.0823}* is verified as the global optimal solution.
- The enumerative searching takes about ***800s*** (CPU i5-4570R 2.7GHz)
- The extended ES optimization takes about ***4.7s***
- With ***(μ, λ) =(150,1000)***, we can always get the above optimal solution. Decreasing the population sizes, for example ***(μ, λ) =(15,100)***, will make the optimization result less consistent.
- As the population converges to the best solution very fast, increasing the number of generations offers less benefits.

Conclusion

- The extended evolutionary strategy is computational efficient and can handle *continuous variables*, *integer variables* and *discrete variables*.
- In extended ES, the discrete variables are mapped to integer indices, and the mutation operation is based on these integer indices.
- The optimization result is sensitive to the population size. A set of sufficient large population sizes need to be used so that the optimization result is consistent and close to the optimal solution.

Reference:

1. Sandgren, E. "Nonlinear integer and discrete programming in mechanical design optimization." Journal of Mechanical Design 112.2 (1990): 223-229.
2. Lee, Cheol W., and Yung C. Shin. "Evolutionary modelling and optimization of grinding processes." International Journal of Production Research 38.12 (2000): 2787-2813.