Lecture 10:
Multi-Objective Optimization Exercises

Multi-Objective Optimization in GOSET

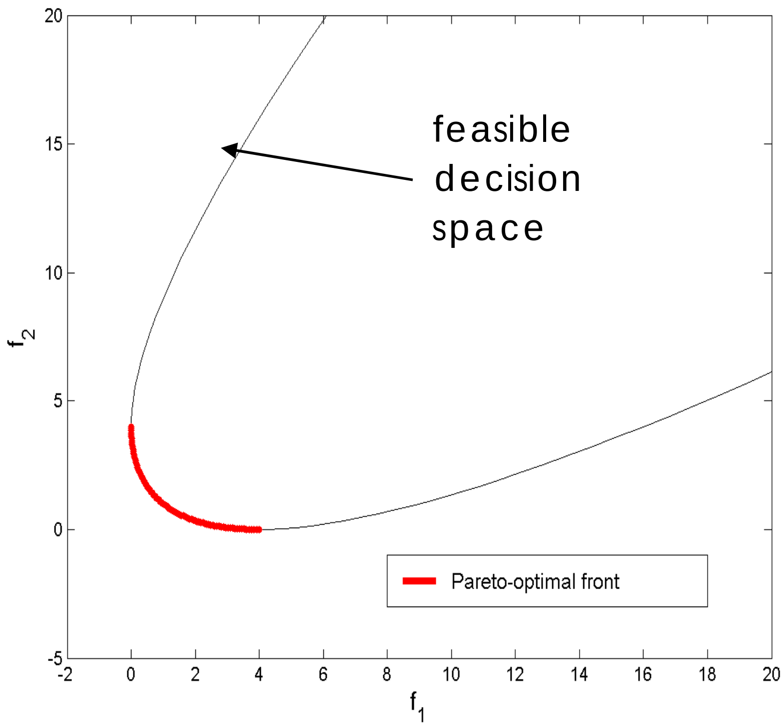
- GOSET employ an **elitist GA** for the multi-objective optimization problem
- **Diversity control algorithms** are also employed to prevent over-crowding of the individuals in a specific region of the solution space
- The non-dominated solutions are identified using the recursive algorithm proposed by **Kung** et al.

Schaffer’s problem

Problem Statement

- Schaffer's problem is a single variable problem with two objectives to be minimized

$$\min\ f_1(x) = x^2$$
$$\min\ f_2(x) = (x-2)^2$$
$$-10 < x < 10$$



- The Schaffer's problem has Pareto optimal solutions $x \in [0, 2]$

Fitness Function

- Two objectives have positive values and they are to be minimized
- The fitness function values are defined to be the negative of the objective function values

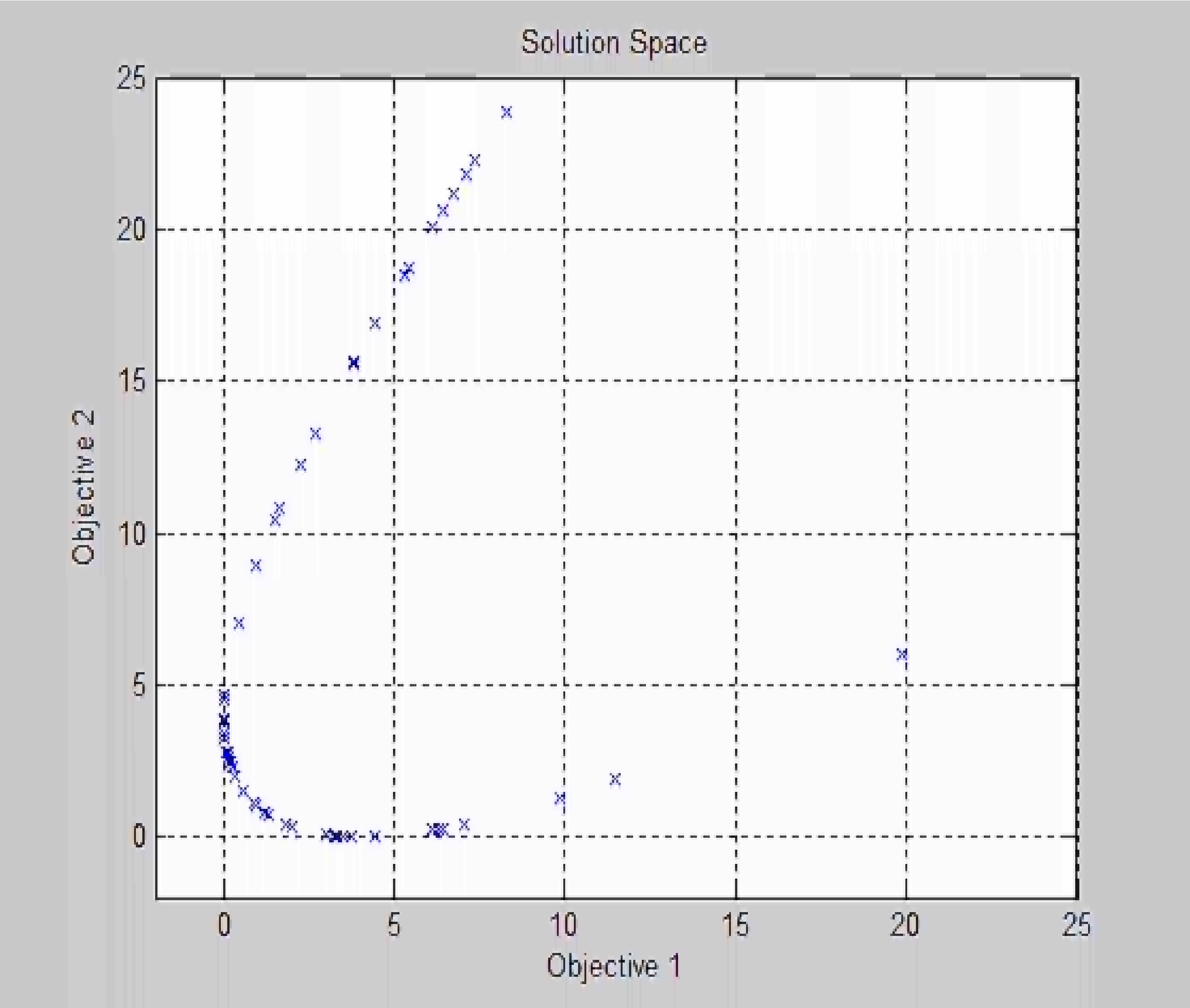
schaffer.m

```
% Schaffer's problem
function [f] = schaffer (x)

f(1,1)= -x(1)^2;
f(2,1)= -(x(1)-2)^2;
```

schaffer_problem.m

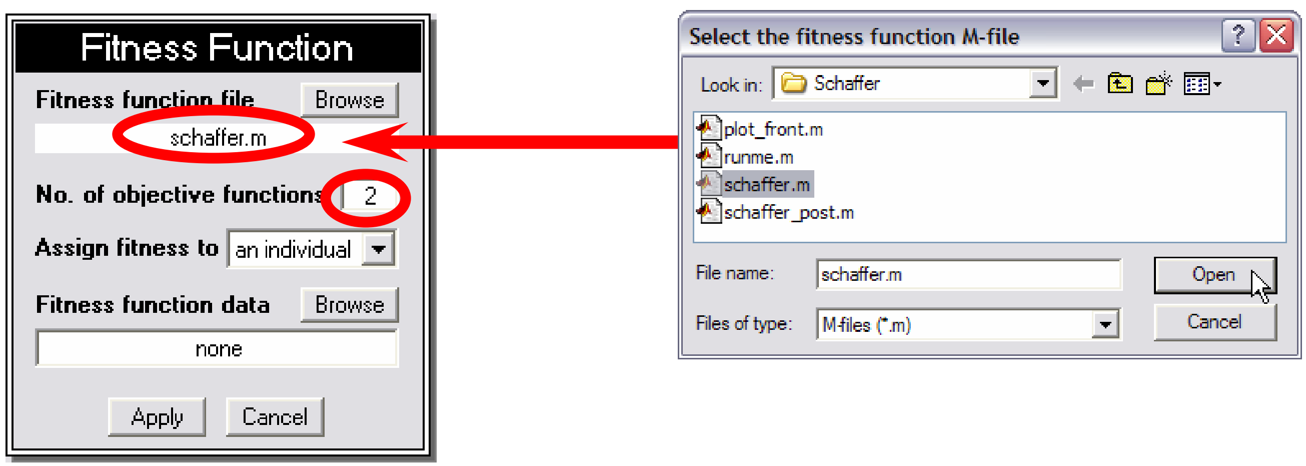
```
% Optimization of Schaffer's Problem

% Initialize the parameters
GAP = gapdefault(2);
GAP.fp_ngen = 20;
GAP.fp_obj = 0;
GAP.sc_alg = 1;
GAP.op_list = [];
GAP.pp_list = [1 2];
GAP.pp_style = [1 1];
GAP.pp_sign = [-1 -1];
GAP.pp_axis = [-2 25 -2 25];

%                  x1
% gene              1
GAP.gd_min  = [  -10   ];
GAP.gd_max  = [   10   ];
GAP.gd_type = [    2   ];
GAP.gd_cid  = [    1   ];

[P,GAS]=gaoptimize(@schaffer_fit,GAP,[],[],[],[]);
```

Pareto Plot (100 Generations)

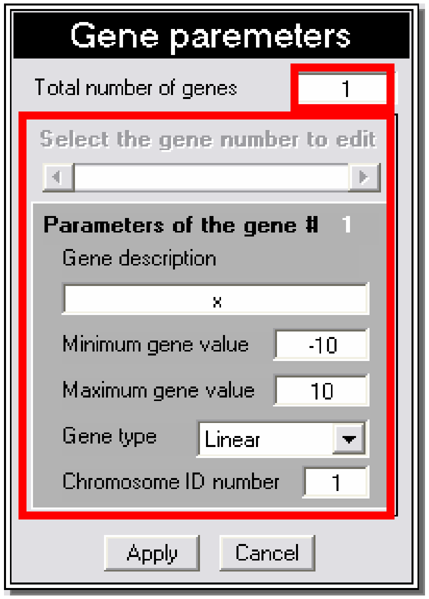


GUI approach

- Define the fitness function
  - In the fitness function window, browse and select the fitness function '`schaffer.m`' defined in the command line approach
  - Schaffer's problem has two objective functions and thus the number of objective functions is set to 2

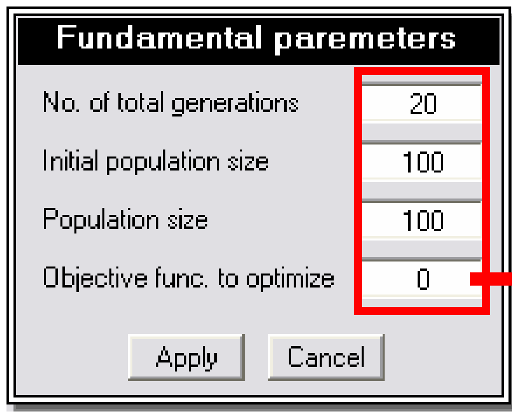


- Define the gene parameters
  - There are only 1 parameter values in this problem, so set the total number of genes to **1**



  - The minimum and maximum gene value for *x* are to **-10** and **10**, '**Linear**' gene type is used

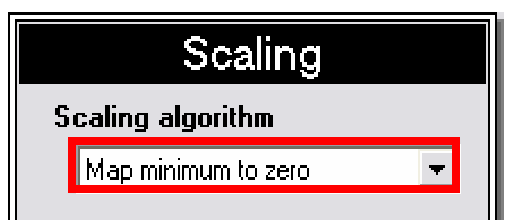
- Define the fundamental parameters
  - In the fundamental parameter input field, set the number of total generations to **20**, the initial population size to **100**, the population size to **100**
  - As there are two objective functions, 'objective function to optimize' is set to **0** then click 'apply'



Caution!!!

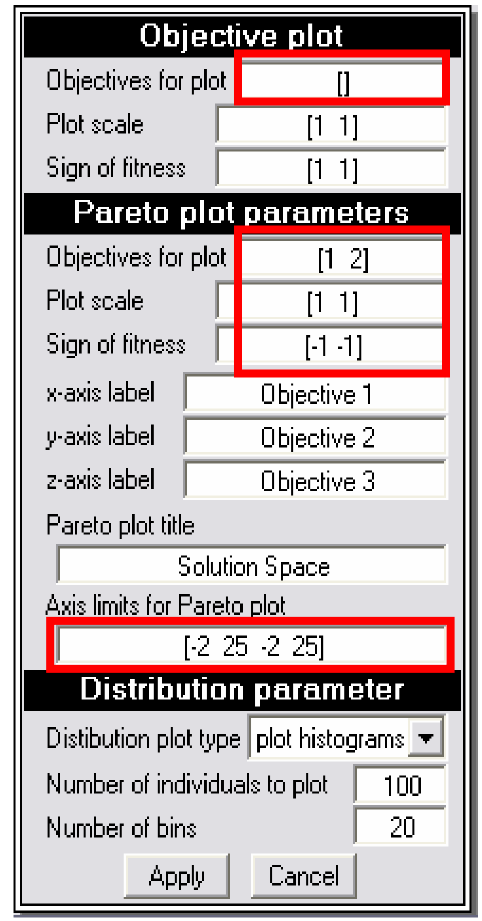
Set 'Objective function to optimize to 0 for multi-objective optimization problem

- Define the scaling parameters
  - The scaling algorithm is set to 'Map minimum to zero'



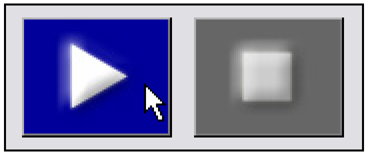
- Define the plotting parameters

  - To display the Pareto plot, ‘Objectives for plot’ for the objective plot is set to ‘[ ]’ and the ‘Objectives for plot’ for the Pareto plot is set to ‘[1 2]’
  - ‘Plot scale’ for the Pareto plot parameters are linear for all objectives
  - Sign of fitness is set to -1 for both objectives
  - Axis limits are defined as [-2 25 -2 25]



- Define the output reporting level
  - For the text and graphical report, check the reporting option is set to '**Text and plot**'

- Start GOSET



Tanaka Problem

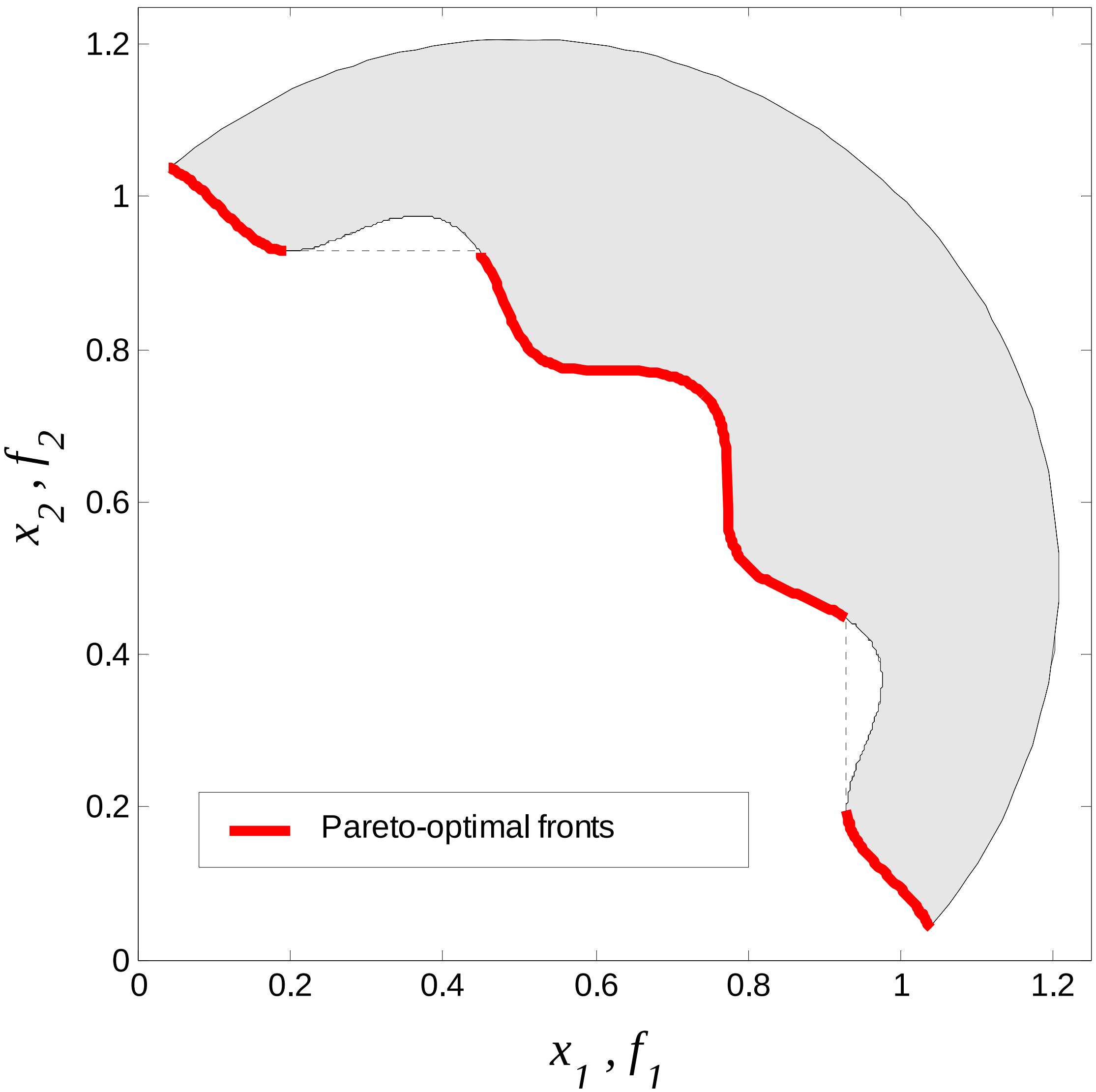
Problem Statement

- Tanaka problem is a constrained optimization problem with two objectives to be minimized

$$
\begin{aligned}
\min\ & f_1(x_1, x_2) = x_1 \\
\min\ & f_2(x_1, x_2) = x_2 \\
\text{subject to } & C_1(x_1, x_2) = x_1^2 + x_2^2 - 1 - 0.1\cos\left(16\arctan\tfrac{x_1}{x_2}\right) \geq 0, \quad 0 \leq x_1 \leq \pi, \\
& C_2(x_1, x_2) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.5, \quad 0 \leq x_2 \leq \pi.
\end{aligned}
$$

- Note the variable space is also the objective space

Tanaka Problem



Fitness Function

- Two objectives have positive values and they are to be minimized
- The fitness function values are defined to be the negative of the objective function values
- Infeasible solutions are assigned with -10 to reduce the chance of surviving

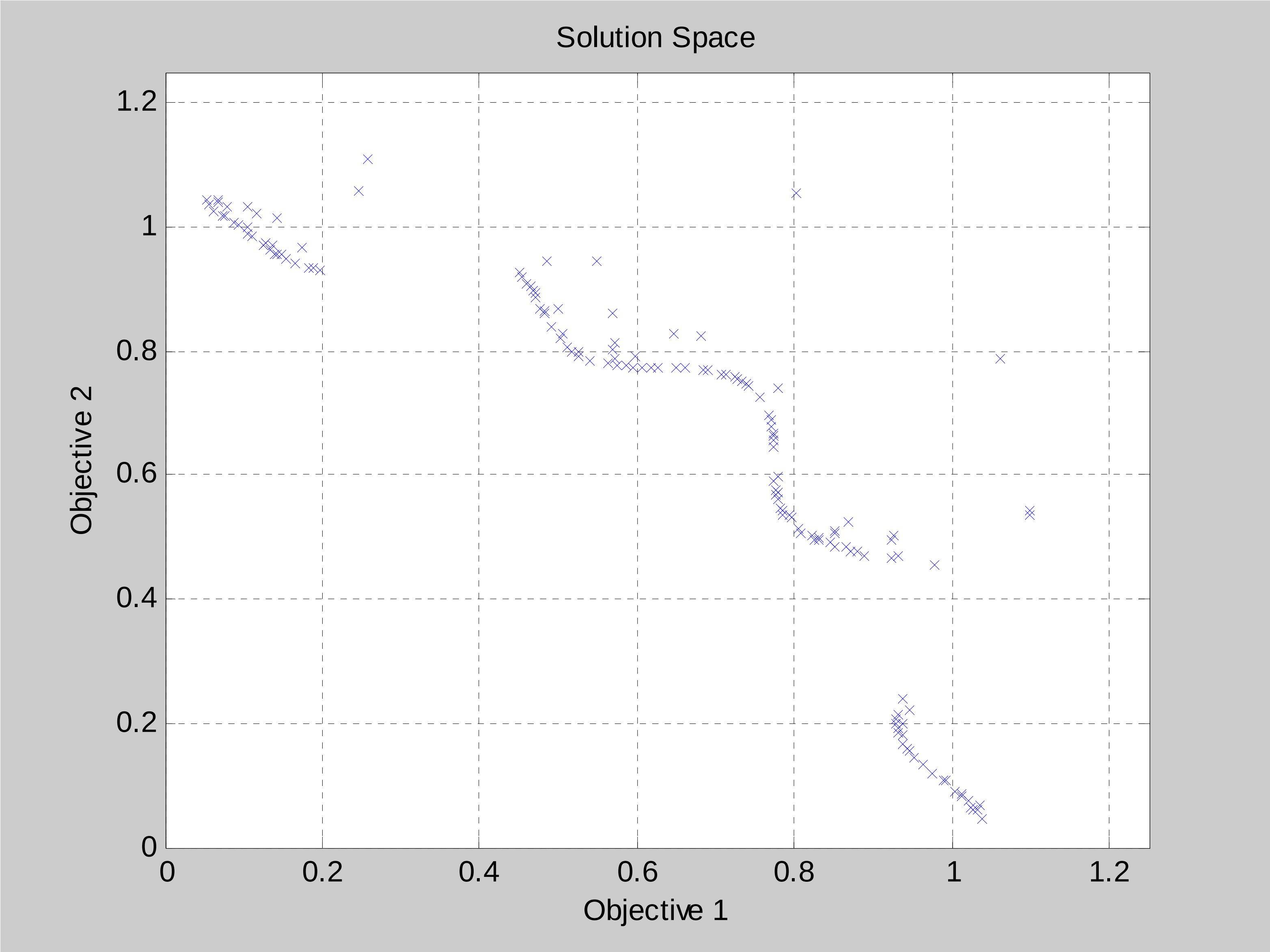
tanaka.m

```
% Tanaka problem (1995)

function [f] = tanaka(x)

C1 = x(1)^2+x(2)^2-1-0.1*cos(16*atan(x(1)/x(2))) >= 0;
C2 = (x(1)-0.5)^2+(x(2)-0.5)^2 <= 0.5;

if C1 & C2
    f(1,1) = -x(1);
    f(2,1) = -x(2);
else
    f(1,1) = -10;
    f(2,1) = -10;

end
```

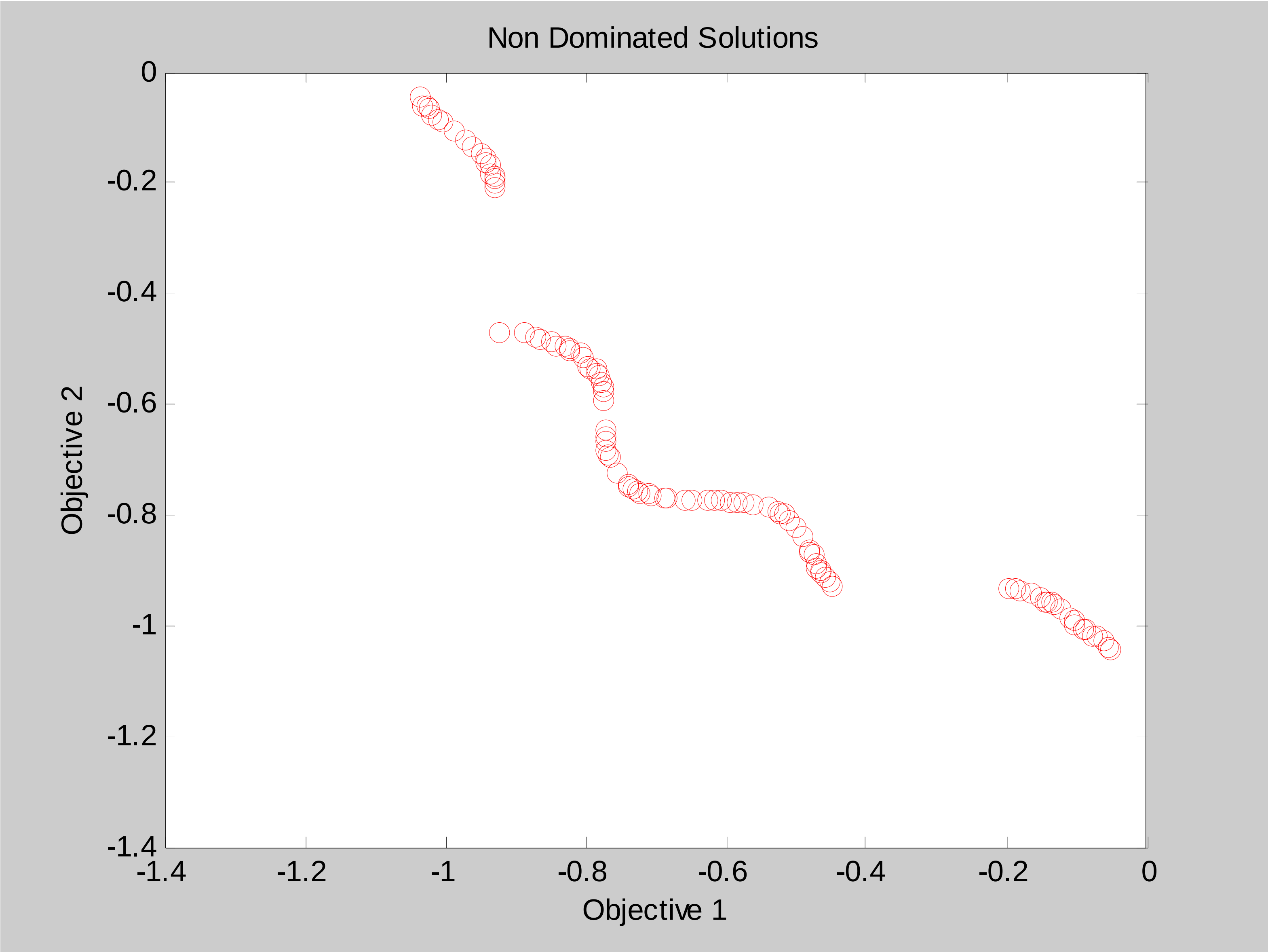
tanaka_problem.m

```
% Initialize the parameters
GAP = gapdefault(2);

GAP.fp_ipop = 200;
GAP.fp_npop = 200;
GAP.fp_ngen = 200;
GAP.fp_obj = 0;
GAP.sc_alg = 6;         % sigma-truncation
GAP.op_list = [];       % list of objectives for objective plots
GAP.pp_list = [1,2];    % list parameters for Pareto plot
GAP.pp_sign = [-1,-1];  % sign of fitness for each objective
GAP.pp_axis=[0 1.25 0 1.25];        % axis limits for Pareto plot

%                 x1      x2
% gene             1       2
GAP.gd_min  = [    0       0    ];
GAP.gd_max  = [   pi      pi    ];
GAP.gd_type = [    2       2    ];
GAP.gd_cid  = [    1       1    ];

% perform optimization
[P,GAS]=gaoptimize(@tanaka,GAP,[],[],[],[]);

% find the nondominated set
nd=nondom(P.mfit,1);
ndi=find(nd==1);
figure(2);
plot(P.mfit(1,ndi),P.mfit(2,ndi),'ro');
title('Non Dominated Solutions');
xlabel('Objective 1');
ylabel('Objective 2');
```

Pareto Plot

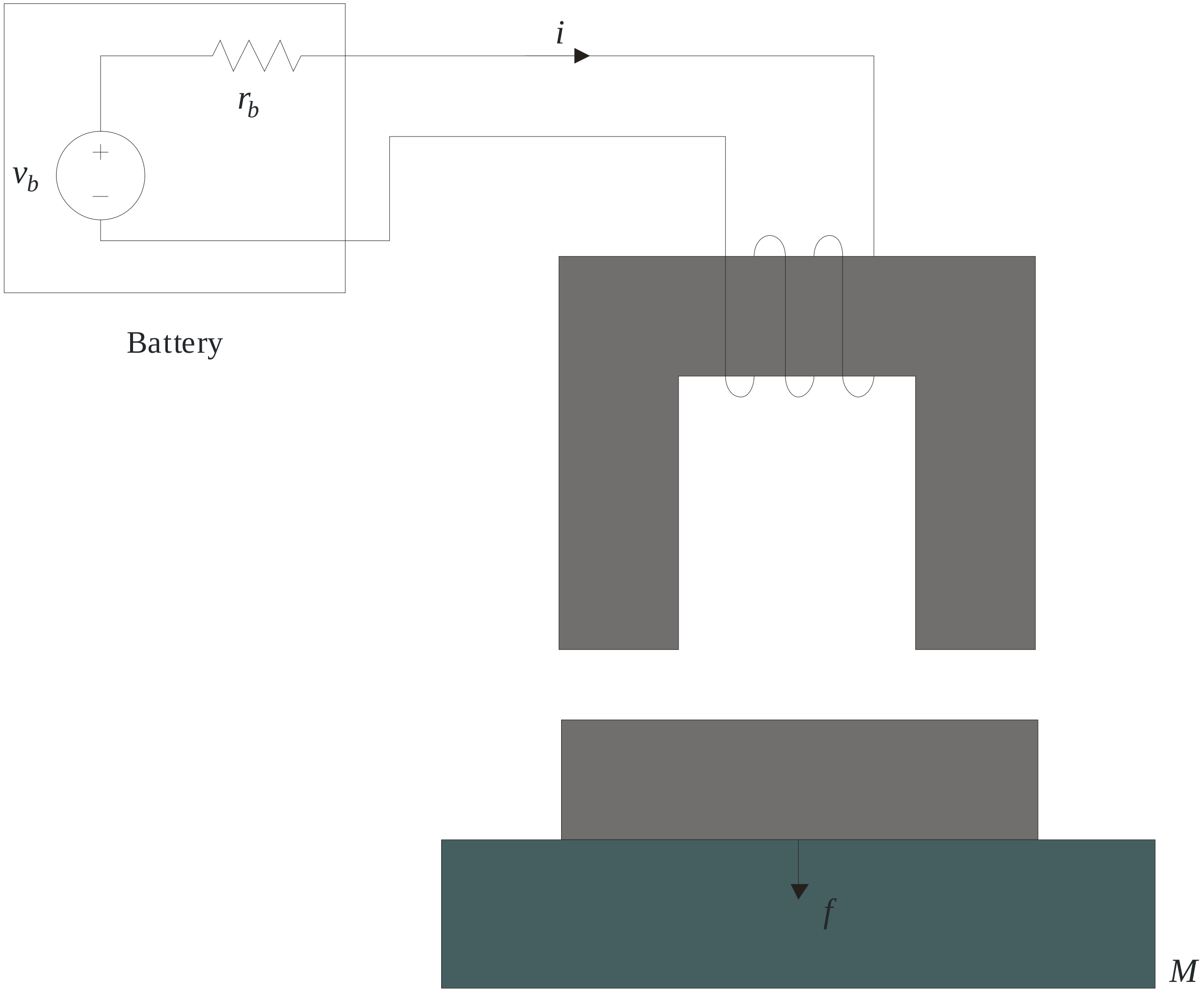


Non Dominated Solutions



Return of the … Electromagnet

Recall Our Electromagnet Design



Multi Objective Fitness Function

$$\mathbf{f} = \begin{cases} \begin{bmatrix} \dfrac{\frac{1}{m_1 + \varepsilon}}{\frac{1}{i + \varepsilon}} \end{bmatrix} & c = 1 \\ \left(\sum_{c=1}^{10} c_i - 10 \right) \begin{bmatrix} 1 \\ 1 \end{bmatrix} & c < 1 \end{cases}$$

New Matlab Files

- Electromagnet_MultiObjective_Fitness.m
- Electromagnet_MultiObjective_Design.m
- Electromagnet_MultiObjective_Design_Review.m

Fitness Function Changes

- Electromagnet_Fitness

```
% compute the fitness
if (cmin == 1)
    fitness = 1/(m1+1e-6);
else
    fitness = sum(c)-length(c);
end
```

- Electromagnet_MultiObjective_Fitness

```
% compute the fitness
if (cmin == 1)
    fitness = [1/(m1+1e-6); 1/(i+1e-6)];
else
    fitness = (sum(c)-length(c))*[1; 1];
end
```

Electromagnet_MultiObjective_Design.m

```
% Electromagnet Design
%
% Written by:
% Ricky Chan for S.D. Sudhoff
% School of Electrical and Computer Engineering
% 1285 Electrical Engineering Building
% West Lafayette, IN 47907-1285
% E-mail: sudhoff@ecn.purdue.edu

% Set Up Population
GAP         = gapdefault(2);
GAP.fp_ngen = 2500;
GAP.fp_ipop = 500;
GAP.fp_npop = 500;
GAP.mc_alg  = 4.0;
GAP.dt_alg  = 3;

% Set Up Reporting
GAP.op_list=[];                 % No objective plot
GAP.pp_list=[1,2];              % Pareto plot
GAP.pp_xl='Inverse Mass';       % Axis labels
GAP.pp_yl='Inverse Current';
GAP.pp_axis=[0.0 2 0.0 1];      % Axis limits

% Set Up Migrations
GAP.mg_nreg=5;       % number of regions
GAP.mg_tmig=100;     % mean time between migrations
GAP.mg_pmig=0.05;    % probability of a individual migrating

% Units
mm=1.0e-3;
cm=1.0e-2;
```

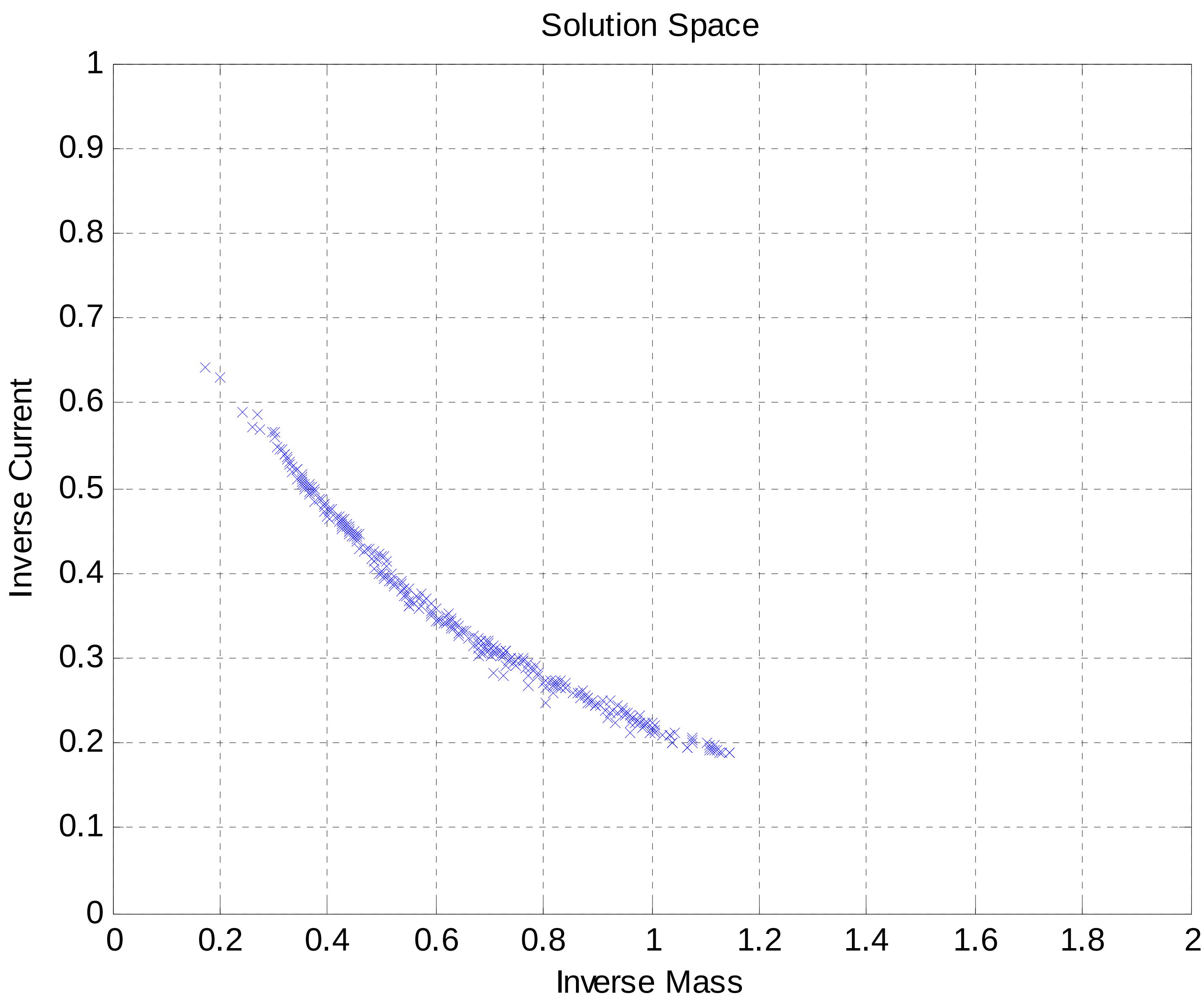
Electromagnet_MultiObjective_Design.m

```
% Problem Requirement Data
Psi.M      = 10;                                          % Mass of object (Kg)
Psi.G      = 9.8;                                         % Gravity (m/s2)
Psi.g      = 0.5*cm;                                      % Gap Width (m)
Psi.rhow   = [    8900;       2701];                      % Wire - Kg/m3
Psi.sigmaw = [    58.0;       35.4]*1e6;                  % Wire - A/Ohm
Psi.jmaxw  = [    7.62;       6.14]*1e6;                  % Wire - A/m2
Psi.descw  = ['Copper  '; 'Aluminum'];                    % Wire - Description
Psi.bmaxm  = [   1.4;              0.7;           1.2];   % Steel - Bsat, T
Psi.rhom   = [7064.1;           8069.4;        7892.1];   % Steel - Density Kg/m3
Psi.myum   = [ 15000;             6000;          3500];   % Steel - Perm., (relative)
Psi.descm  = ['Microsil    '; ...
              'Superperm 80'; ...
              'Superperm 49'];                            % Steel - Description
Psi.imax   = 6;                                           % Maximum current, A
Psi.lmax   = 10e-2;                                       % Maximum length
Psi.pfmax  = 0.7;                                         % Maximum packing factor
Psi.vb     = 12;                                          % Battery Voltage
Psi.rb     = 0.5;                                         % Battery Resistance

% Genetic Mapping
%                mw   ac    N    mm    ws    rww    ds    rdw   we    rwi  rwb   d
GAP.gd_min  = [  1   1e-8   10    1   1*cm   0.1   1*mm   0.1  1*mm   0.2  0.2  1*mm];
GAP.gd_max  = [  2   1e-4  1e3    3  10*cm   1.0  20*cm   1.0  5*cm   2.0  2.0 10*cm];
GAP.gd_type = [  1    3     3     1    3      3     3      3     3     3    3    3  ];
GAP.gd_cid  = [  1    1     1     1    1      1     1      1     1     1    1    1  ];

% Solve Problem
[fP,GAS]= gaoptimize(@Electromagnet_MultiObjective_Fitness,GAP,Psi,[],[],[]);

% Save run
s=input('Type 1 to Save Run ');
if (s==1)
   save sample_multi_objective_run
end
```

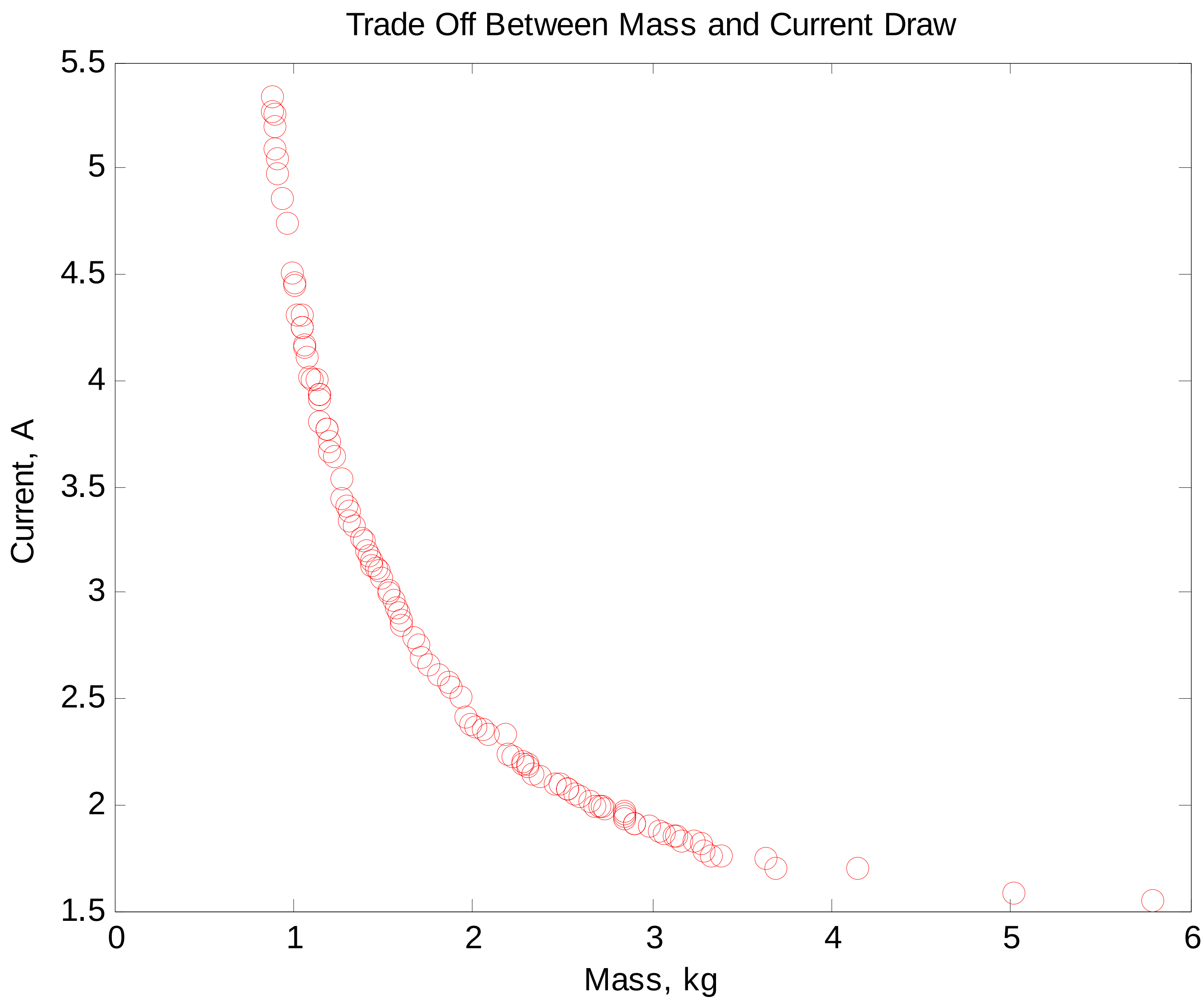
Electromagnet_MultiObjective_Design_Review.m

```
% Electromagnet Design Review
% Illustrates the evolution of the electromagnet design
%
% Written by:
% S.D. Sudhoff
% School of Electrical and Computer Engineering
% 1285 Electrical Engineering Building
% West Lafayette, IN 47907-1285
% E-mail: sudhoff@ecn.purdue.edu

clear all

load sample_multi_objective_run

% Plot Pareto Optimal Front in Objective Space
non_dominated_status=nondom(fP.mfit,1);
non_dominated_indices=find(non_dominated_status==1);
Mass=1./fP.mfit(1,non_dominated_indices)-1e-6;
Current=1./fP.mfit(2,non_dominated_indices)-1e-6;
figure(2)
plot(Mass,Current,'ro');
xlabel('Mass, kg');
ylabel('Current, A');
title('Trade Off Between Mass and Current Draw');

% Minimum Mass Design
mmd=GAS.bestgenes(:,GAS.cg,1);
Electromagnet_MultiObjective_Fitness(mmd,Psi,3);

% Minimum Current Design
mcd=GAS.bestgenes(:,GAS.cg,2);
Electromagnet_MultiObjective_Fitness(mcd,Psi,5);
```

Pareto Plot


Solution Space
Inverse Current
Inverse Mass
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2

Design Space

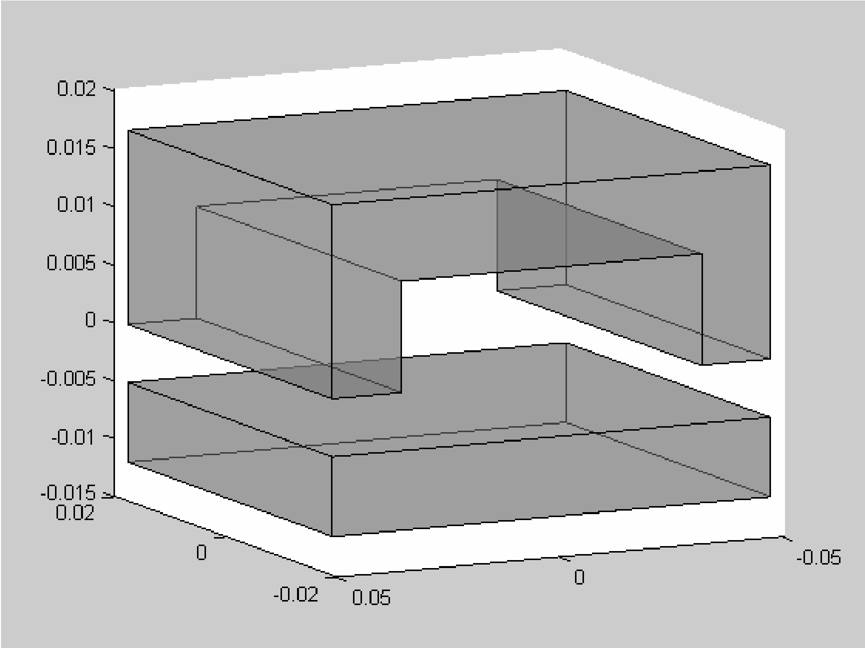


Design Extremes

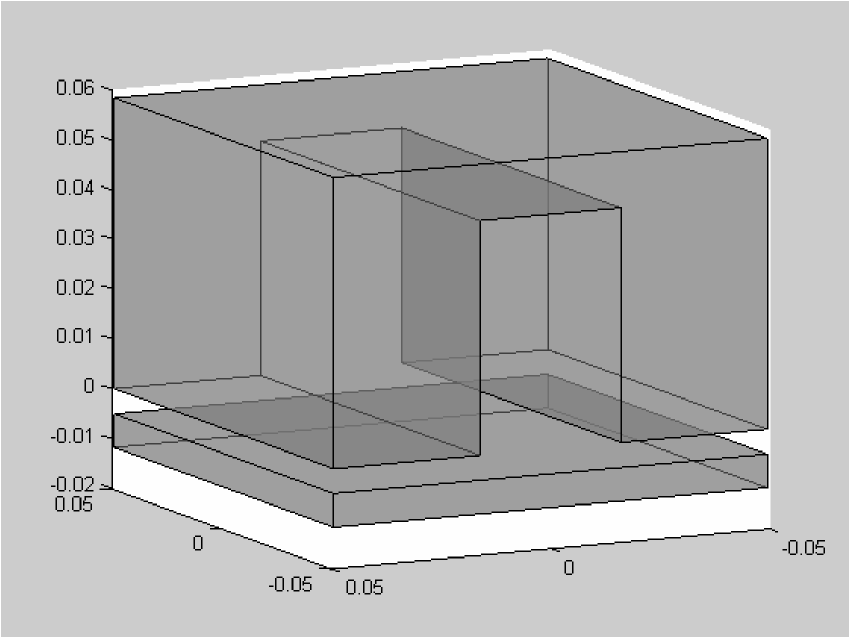
- Design 1
  - Mass: 0.874 kg
  - Current: 5.33 A
- Design 2
  - Mass: 5.79 kg
  - Current: 1.56 A

Design Extremes

Design 1

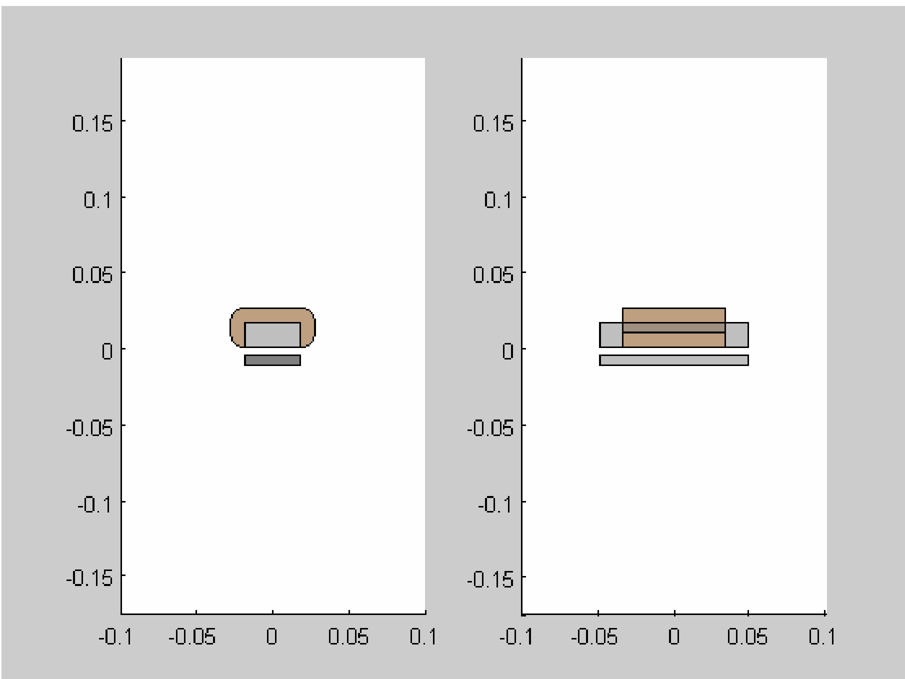


Design 2



Design Extremes

Design 1



Design 2