NNF

Optimize for all 7 parameters: {delta, Amax, Vmax, Ka, alpha, Km, beta\_max}

1. Relative amp of Ptot > 0.25

2. max(Ptot) is minimized s.t Kd is maximized

~~3. BMAL (At) average level matches SNF model~~

4. Relative amp of BMAL > 0.2

~~5. Period of Ptot matches SNF~~

6. max(Rev) < 10 (max(Ror)<5 for PNF)

Method: simulated annealing

Options: 'TemperatureFcn','temperatureboltz', 'AnnealingFcn','annealingboltz',

Cost function:

penalty = 10;

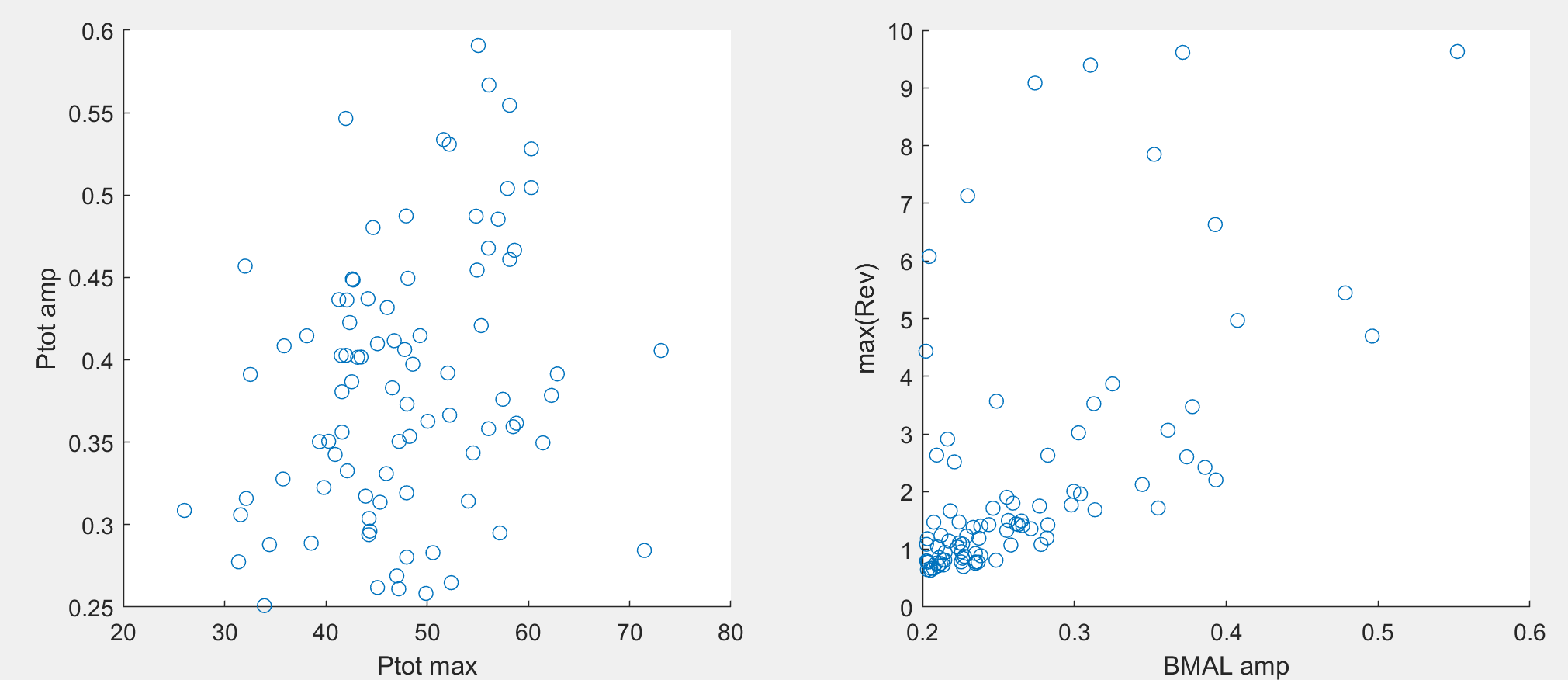
cost\_item = [penalty\*(BMAL\_amp./BMAL\_amp\_sim+(BMAL\_amp\_sim/BMAL\_amp-1)^2)\*(BMAL\_amp\_sim < BMAL\_amp)+0.01; ...% amp>0.2

penalty\*(maxRev\_sim/Rev\_max-1)^2\*(maxRev\_sim > Rev\_max)+0.01; ... % max(Rev)<10

Ptot\_max\_sim/max\_Ptot\_SNF; ... % minimize max(Ptot)

penalty\*(Ptot\_amp/Ptot\_amp\_sim+(Ptot\_amp\_sim/Ptot\_amp-1)^2)\*(Ptot\_amp\_sim < Ptot\_amp)+0.01]; % amp > 0.25

cost = sum(cost\_item);

Results of 128 runs (‘NNF\_fit\_batch.mat’):

Best set:

[delta, AMAX, VMAX, KA, alpha, Km, beta\_max]= [0.3173 9.9232 9.8926 2.5433 12.7306 1.1993 3.0514]

Ptot\_max\_sim = 26.0253

Ptot\_amp\_sim = 0.3086

BMAL\_amp\_sim = 0.2165

maxRev\_sim = 2.9117

cost = 0.3770

cost\_item = [0.0100, 0.0100, 0.3470, 0.0100]

SNF

Optimize for all 5 parameters: {At, Ka, alpha, Km, beta\_max}

1. Relative amp of Ptot > 0.25

2. max(Ptot) is minimized s.t Kd is maximized

Method 1: simulated annealing,

Options: 'TemperatureFcn','temperatureboltz', 'AnnealingFcn','annealingboltz'

Cost function:

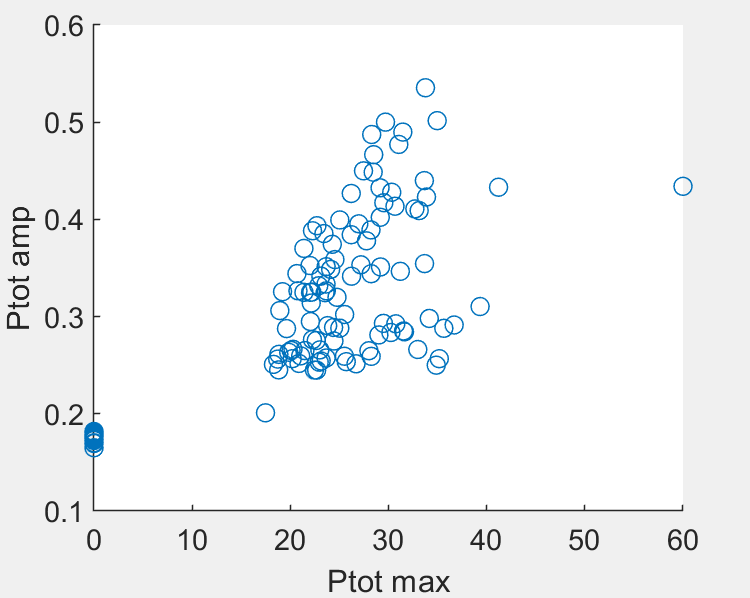
penalty = 1;

cost\_item = [Ptot\_max\_sim/max\_Ptot\_SNF; ... % minimize max(Ptot)

penalty\*((Ptot\_amp/Ptot\_amp\_sim+(Ptot\_amp\_sim/Ptot\_amp-1)^2)\*(Ptot\_amp\_sim < Ptot\_amp)+1\*(Ptot\_amp\_sim >= Ptot\_amp))]; % amp > 0.25

cost = sum(cost\_item);

Results for 128 runs:



Best fit set:

[At, KA, alpha, Km, beta\_max] = [3.6335 1.2782 5.4937 1.0040 2.5484]

Ptot\_max\_sim = 18.2958

Ptot\_amp\_sim = 0.2508

cost = 1.2439

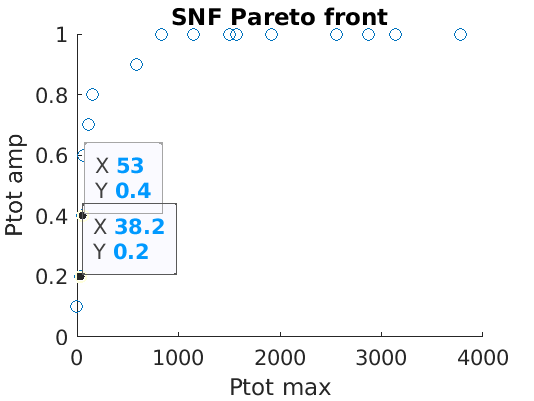
cost\_item = [0.2439 1.0000];

Method 2: Genetic algorithm multiobjective

cost\_item = [Ptot\_max\_sim; ... % minimize max(Ptot)

-Ptot\_amp\_sim]; % maximize Ptot amp

Pareto front:



Remarks: This algorithm does not generate results better than simulated annealing, but clearly shows that small Ptot\_max incurs small Ptot\_amp.