Methodology (Project 3)

Michael Nelson

Contents

Introduction

In this project, we use the weighted-sum and epsilon-constraint methods to solve four biobjective optimization problems.

1 Non-convex BOP with non-connected Pareto set

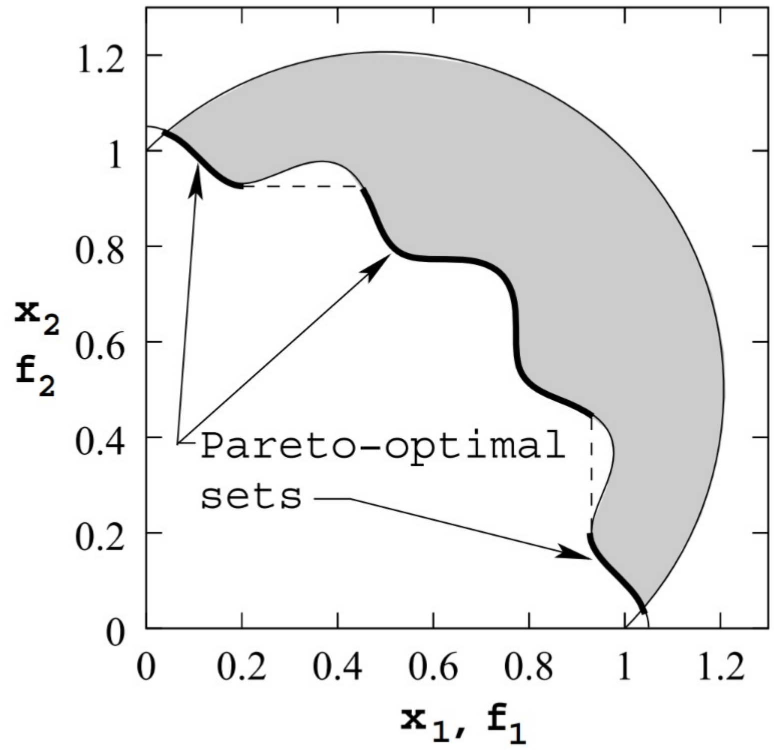
For this problem, let

$$\begin{aligned}
f_1(\boldsymbol{x}) &= x_1 \\
f_2(\boldsymbol{x}) &= x_2 \\
c_1(\boldsymbol{x}) &= 1 + 0.1\cos\left(16\arctan\frac{x_1}{x_2}\right) - x_1^2 - x_2^2 \\
c_2(\boldsymbol{x}) &= (x_1 - 0.5)^2 + (x_2 - 0.5)^2 - 0.5 \\
X &= \{\boldsymbol{x} \in [0,\pi]^2 \subseteq \mathbb{R}^2 \mid \boldsymbol{c}(\boldsymbol{x}) \leqq \boldsymbol{0}\}.
\end{aligned}$$

We consider the following BOP:

$$\begin{aligned}
&\text{minimize} \quad [f_1(x), f_2(x)] \\
&\text{subject to} \quad \boldsymbol{x} \in X.
\end{aligned} \tag{1}$$

This problem is considered as a test problem in [DPM02]. Since $x_1 = f_1$ and $x_2 = f_2$, the feasible and objective space coincide and is shown in the figure below:



In particular, the feasible set X is not convex set and the Pareto front Y_N is disconnected, so this is a non-convex non-connected Pareto set BOP.

1.1 Weighted-Sum Method

We first solve (1) using the weighted-sum method. In other words, we solve the single objective problem

$$\begin{aligned} \text{minimize} \quad & F_w(\boldsymbol{x}) \\ \text{subject to} \quad & \boldsymbol{x} \in X \\ & 0 \leq w \leq 1 \end{aligned} \tag{2}$$

where we set

$$F_w(\boldsymbol{x}) = wf_1(\boldsymbol{x}) + (1-w)f_2(\boldsymbol{x})$$

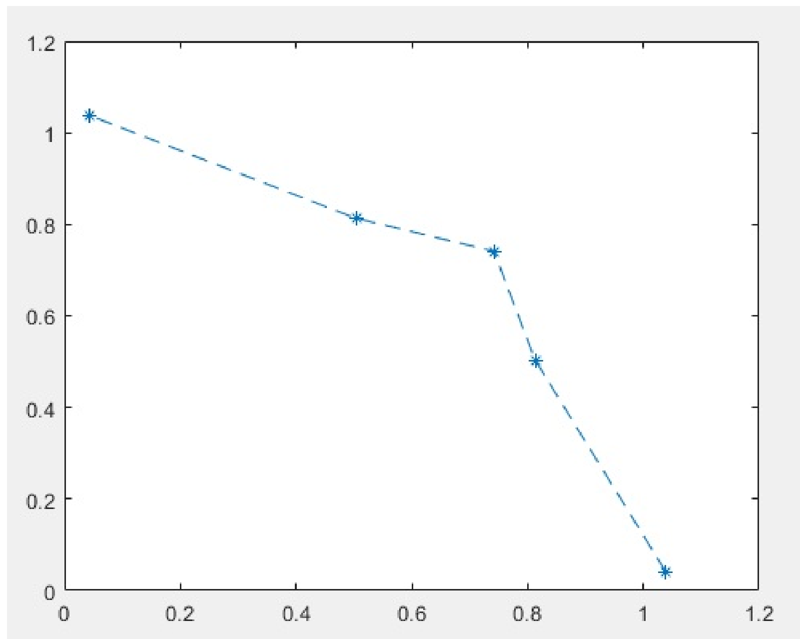
We will find optimals solutions to (2) using the MATLAB function x = fmincon(fun,xo,lb,ub,nonlcon) where the programming solver assumes that the programming has the form

$$\begin{aligned} \text{minimize} \quad & F_w(\boldsymbol{x}) \\ \text{subject to} \quad & \boldsymbol{c}(\boldsymbol{x}) \leqq \boldsymbol{0} \\ & \boldsymbol{lb} \leqq \boldsymbol{x} \leqq \boldsymbol{ub}. \end{aligned}$$

The function starts at an initial guess $\boldsymbol{x}^0$ and attempts to find a *local* optimal solution $\widehat{\boldsymbol{x}}$. Further analysis will be needed in order to determine whether or not $\widehat{\boldsymbol{x}}$ is *global* optimal solution. Let us begin by using $\boldsymbol{x}^0 = (0.8, 0.8)^\top$ to be our initial guess. After setting up the correct MATLAB code (which is given in the Appendix), MATLAB gives us the following table:

w	exit flag	x1	x2	f1	f2
0	1	1.0384	0.041665	1.0384	0.041665
0.1	1	1.0384	0.041665	1.0384	0.041665
0.2	1	1.0384	0.041664	1.0384	0.041664
0.3	1	1.0384	0.041665	1.0384	0.041665
0.4	1	0.81296	0.50376	0.81296	0.50376
0.5	1	0.74162	0.74162	0.74162	0.74162
0.6	1	0.50376	0.81296	0.50376	0.81296
0.7	2	0.041665	1.0384	0.041665	1.0384
0.8	1	0.041664	1.0384	0.041664	1.0384
0.9	1	0.041665	1.0384	0.041665	1.0384
1	1	0.041665	1.0384	0.041665	1.0384

Let us explain what this table is telling us since we will be using the same format throughout the rest of the paper. We are using the weights $w = 0, 0.1, 0.2, \ldots, 1$ which is given in the first column in the table above. Consider the row corresponding to the weight $w = 0.5$. The (approximate) local optimal solution that we found corresponding to the weight $w = 0.5$ is given by $\widehat{\boldsymbol{x}} = (0.74162, 0.74162)^\top$. The f_1-value of $\widehat{\boldsymbol{x}}$ is $f_1(\widehat{\boldsymbol{x}}) = 0.74162$ and the f_2-value of $\widehat{\boldsymbol{x}}$ is $f_2(\widehat{\boldsymbol{x}}) = 0.74162$. Finally, the exit flag value is equal to 1, which tells us that $\widehat{\boldsymbol{x}}$ is a very good approximate to the local optimal solution. In general, positive exit flags correspond to successful outcomes, negative exit flags correspond to unsuccessful outcomes, and zero exit flag corresponds to the solver being halted by exceeding an iteration limit or limit on the number of function evaluations. This explains everything about the row corresponding to $w = 0.5$. The other rows in the table have similar interpretations as well. We now plot the objective values corresponding to the local optimal solutions that we found above:



In order to determine which of these points are guaranteed to be Pareto points, we will appeal to Proposition 3.9 on page 71 in the course text book which we will recall here (keeping the notation as given in the book):

Proposition 1.1. *Suppose $\widehat{\boldsymbol{x}}$ is a (global) optimal solution of the weighted sum optimization problem*

$$\min_{\boldsymbol{x} \in X} \sum_{k=1}^{p} w_k f_k(\boldsymbol{x}) \tag{3}$$

with $\boldsymbol{w} \in \mathbb{R}^p_{\geq}$. Then the following statement hold:

1. *If $\boldsymbol{w} \in \mathbb{R}^p_{\geq}$, then $\widehat{\boldsymbol{x}}$ is weakly efficient;*
2. *If $\boldsymbol{w} \in \mathbb{R}^p_{>}$ and $Y := \boldsymbol{f}(X)$ is $\mathbb{R}^p_{\geqq}$-convex, then $\widehat{\boldsymbol{x}}$ is efficient (so $\widehat{\boldsymbol{y}} = \boldsymbol{f}(\widehat{\boldsymbol{x}})$ is a Pareto point or a non-dominated point);*
3. *If $\boldsymbol{w} \in \mathbb{R}^p_{\geq}$, Y is $\mathbb{R}^p_{\geqq}$-convex, and $\widehat{\boldsymbol{x}}$ is the* unique *optimal solution of (3) $\widehat{\boldsymbol{x}}$ is strictly efficient.*

Recall that this proposition only applies to *global* optimal solutions, and the MATLAB function that we used only calculates local optimal solutions. In fact, it's easy to check that if $0 \leq w < 1/2$ then there's a unique global optimal solution (approximately) at $\boldsymbol{a} := (1.0384, 0.041665)^\top$, if $w = 1/2$ then there are two global optimal solutions (approximately) at $\boldsymbol{a}$ and $\boldsymbol{b} := (0.041665, 1.0384)^\top$, and finally if $1/2 < w \leq 1$, then there's a unique global optimal solution (approximately) at $\boldsymbol{b}$. In any case, Y is not $\mathbb{R}^p_{\geqq}$-convex, and so at the moment, we can only deduce that these global optimal solutions are merely weakly efficient (we shall see that they are indeed efficient using the epsilon-constraint method).

1.2 Epsilon-Constraint Method

Next we solve (1) using the epsilon-constraint method. First let us recall Proposition 4.3, Proposition 4.4, and Theorem 4.5 on pages 99-100 in the course text book which we summarize in the proposition below (keeping the notation the same as given in the book):

Proposition 1.2. *For each $j = 1, \dots, p$, we consider the ε-constraint problem:*

$$\begin{array}{ll} \textit{minimize} & f_j(\boldsymbol{x}) \\ \textit{subject to} & \boldsymbol{x} \in X \\ & f_k(\boldsymbol{x}) \le \varepsilon_k \quad k = 1, \dots, p \quad k \neq j \end{array} \tag{4}$$

where $\boldsymbol{\varepsilon} \in \mathbb{R}^p$. The following statements hold:

1. *If $\widehat{\boldsymbol{x}}$ is an optimal solution of (4) for some j, then $\widehat{\boldsymbol{x}}$ is weakly efficient;*

2. *If $\widehat{\boldsymbol{x}}$ is the unique optimal solution to (4) for some j, then $\widehat{\boldsymbol{x}}$ is strictly efficient (and hence $\widehat{\boldsymbol{x}}$ is efficient);*

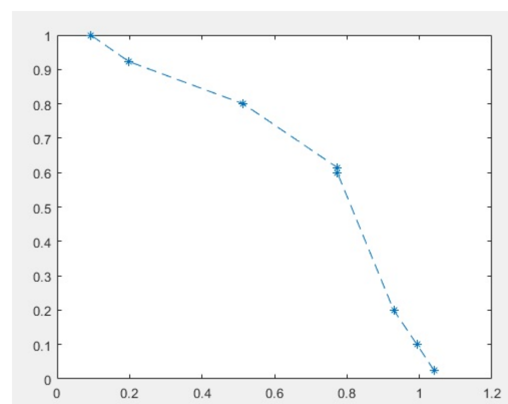
3. *$\widehat{\boldsymbol{x}}$ is efficient if and only if there exists an $\widehat{\boldsymbol{\varepsilon}}$ such that $\widehat{\boldsymbol{x}}$ is an optimal solution of (4) for all j.*

In this case, we only need to consider the following epsilon-constraint problem

$$\begin{array}{ll} \text{minimize} & f_1(\boldsymbol{x}) \\ \text{subject to} & \boldsymbol{x} \in X \\ & f_2(\boldsymbol{x}) - \varepsilon \le 0 \end{array} \tag{5}$$

where $0.1 \le \varepsilon \le 1$. Indeed, it is straightforward to check that (5) will have a unique optimal solution for each such ε and so we can apply part 2 of Proposition (1.2). Furthermore, this unique optimal solution will be the only *local* optimal solution, so we can use the MATLAB function fmincon to obtain the optimal solutions to (5) for each such ε. MATLAB produces the following table and plot for this method:

e	exit flag	x1	x2	f1	f2
0	-2	1.0406	0.023927	1.0406	0.023927
0.1	1	0.99324	0.1	0.99324	0.1
0.2	1	0.92905	0.19958	0.92905	0.19958
0.3	1	0.92905	0.19963	0.92905	0.19963
0.4	1	0.92905	0.19963	0.92905	0.19963
0.5	1	0.92905	0.19963	0.92905	0.19963
0.6	1	0.77315	0.6	0.77315	0.6
0.7	1	0.77308	0.61473	0.77308	0.61473
0.8	1	0.51394	0.8	0.51394	0.8
0.9	-2	0.1981	0.92292	0.1981	0.92292
1	1	0.093027	1	0.093027	1



We found much more success using the epsilon-constraint method over the weighted-sum method for this problem thanks to Proposition (1.2) which tells us that these points are indeed efficient.

2 Non-convex BOP with connected Pareto set

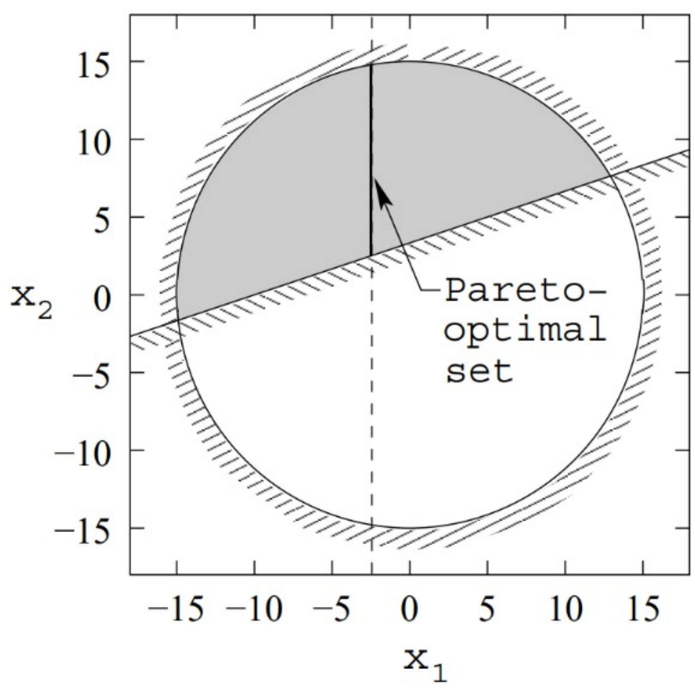
For this problem, let

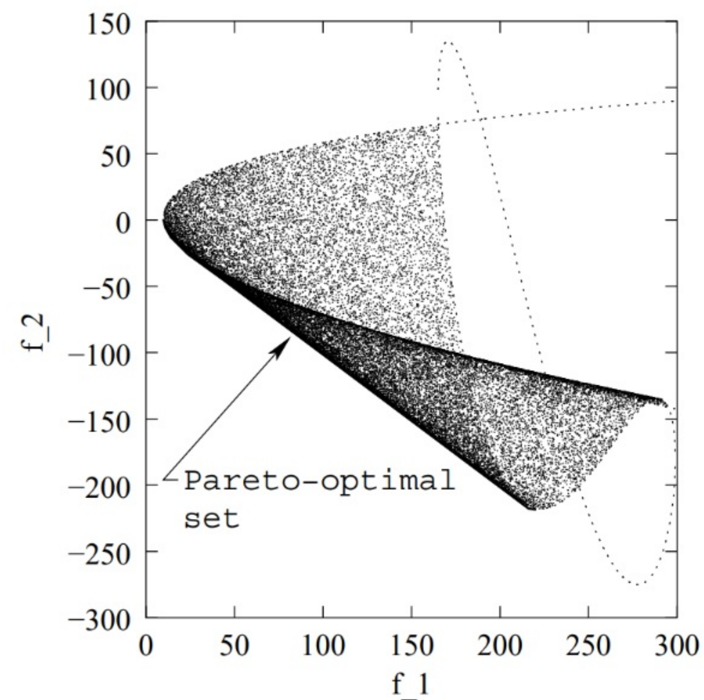
$$\begin{aligned} f_1(\boldsymbol{x}) &= 2 + (x_1 - 2)^2 + (x_2 - 1)^2 \\ f_2(\boldsymbol{x}) &= 9x_1 - (x_2 - 1)^2 \\ c_1(\boldsymbol{x}) &= x_1^2 + x_2^2 - 225 \\ c_2(\boldsymbol{x}) &= x_1 - 3x_2 + 10 \\ X &= \{\boldsymbol{x} \in [-20, 20]^2 \subseteq \mathbb{R}^2 \mid \boldsymbol{c}(\boldsymbol{x}) \leqq 0\}. \end{aligned}$$

we consider the following BOP:

$$\begin{array}{ll} \text{minimize} & [f_1(x), f_2(x)] \\ \text{subject to} & \boldsymbol{x} \in X. \end{array} \tag{6}$$

This problem is considered as a test problem in [DPM02-1]. The feasible set X as well as the objective outcome set $Y = \boldsymbol{f}(X)$ is illustrated below:





Note that the Pareto front Y_N is connected. Also note that the Hessian of f_1 is $\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$ and the Hessian of f_2 is $\begin{pmatrix} 0 & 0 \\ 0 & -2 \end{pmatrix}$. In particular, f_1 is strictly convex, however f_2 is not convex (it is concave instead). Thus this is a non-convex BOP whose Pareto front is connected.

2.1 Weighted-Sum Method

We first solve (6) using the weighted-sum method. In other words, we solve the single objective problem

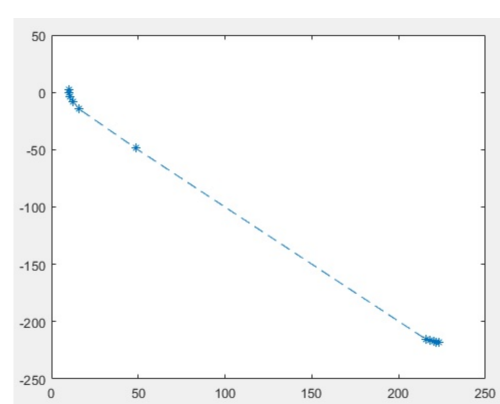
$$\begin{aligned} \text{minimize} \quad & F_w(\boldsymbol{x}) \\ \text{subject to} \quad & \boldsymbol{x} \in X \\ & 0 \leq w \leq 1 \end{aligned} \tag{7}$$

where we set

$$F_w(\boldsymbol{x}) = w f_1(\boldsymbol{x}) + (1 - w) f_2(\boldsymbol{x}).$$

Just like in the previous problem, we will find local optimal solutions to (7) (corresponding to each w) using the MATLAB function fmincon (all of our MATLAB can be found in the Appendix). MATLAB produces the following table and plot:

w	exit flag	x1	x2	f1	f2
0	1	-4.841	14.197	222.97	-217.74
0.1	1	-4.5615	14.29	221.67	-217.67
0.2	1	-4.2199	14.394	220.09	-217.38
0.3	1	-3.7922	14.513	218.14	-216.72
0.4	1	-3.2404	14.646	215.67	-215.37
0.5	1	-2.5	6.1226	48.491	-48.741
0.6	1	-1.2143	2.9286	16.051	-14.648
0.7	1	-0.35075	3.2164	12.439	-8.0692
0.8	1	0.26923	3.4231	10.867	-3.4482
0.9	1	0.73595	3.5787	10.247	-0.025852
1	1	1.1	3.7	10.1	2.61



Note however that all we can say is that these give weakly efficient solutions (with the exception of the point corresponding to weight $w = 0.5$). Indeed, Y is not $\mathbb{R}^2_{\geqq}$-convex, and so we cannot apply the full strength of Proposition (1.1).

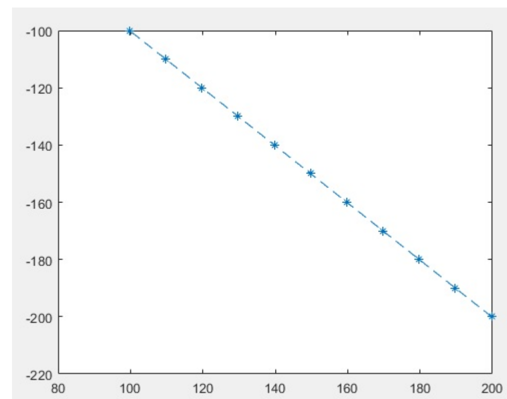
2.2 Epsilon-Constraint Method

Next we solve (6) using the epsilon-constraint method. In this case, we only need to consider the following epsilon-constraint problem

$$\begin{aligned} \text{minimize} \quad & f_2(\boldsymbol{x}) \\ \text{subject to} \quad & \boldsymbol{x} \in X \\ & f_1(\boldsymbol{x}) - \varepsilon \leq 0 \end{aligned} \tag{8}$$

where $-400 \leq \varepsilon \leq -100$. Then f_2 is convex in the new feasible region remains convex, so this just becomes a convex problem. Furthermore one can show that (8) has a unique (local and global) optimal solution in this case, and so we can apply part 2 of Proposition (1.2). MATLAB produces the following table and plot:

e	exit flag	x1	x2	f1	f2
-200	1	-2.5	14.323	199.75	-200
-190	1	-2.5	13.942	189.75	-190
-180	1	-2.5	13.55	179.75	-180
-170	1	-2.5	13.145	169.75	-170
-160	1	-2.5	12.726	159.75	-160
-150	1	-2.5	12.292	149.75	-150
-140	1	-2.5	11.84	139.75	-140
-130	1	-2.5	11.368	129.75	-130
-120	1	-2.5	10.874	119.75	-120
-110	1	-2.5	10.354	109.75	-110
-100	1	-2.5	9.8034	99.75	-100



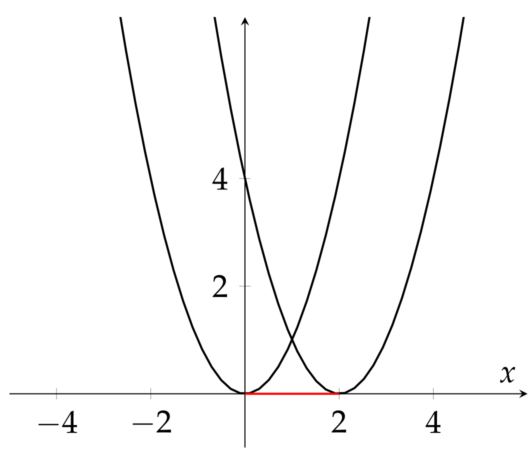
The epsilon-constraint method prevails over the the weighted-sum method yet again!

3 Convex BOP with connected Pareto set

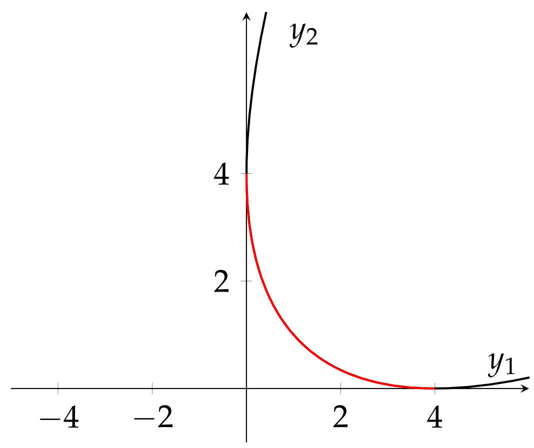
For this problem, let $f_1(x) = x^2$, let $f_2(x) = (x-2)^2$, and let $X = [-5, 5]$. We consider the following BOP:

$$\begin{aligned} \text{minimize} \quad & [f_1(x), f_2(x)] \\ \text{subject to} \quad & x \in X. \end{aligned} \tag{9}$$

This problem is considered as a test problem in [HHBW]. In the image below, we draw the graphs of f_1 and f_2, and we also draw the efficient set $X_E = [0, 2]$ (shaded in red):



Next we draw the outcome set $Y = \boldsymbol{f}(X)$ together with the Pareto front Y_N (shaded in red):



3.1 Weighted-Sum Method

We first solve (9) using the weighted-sum method. In other words, we solve the single objective problem

$$\begin{array}{ll} \text{minimize} & F_w(x) \\ \text{subject to} & x \in X \end{array} \tag{10}$$

where $0 \leq w \leq 1$ and where

$$F_w(x) = wf_1(x) + (1-w)f_2(x)$$

We can easily solve this BOP by hand. Indeed, note that

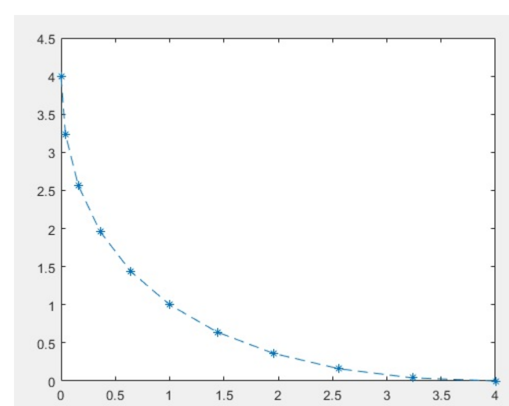
$$F_w'(x) = 2(2w + x - 2).$$

In particular, F_w has exactly one critical point: $c_w = 2(1-w)$. Furthermore observe that $F_w''(x) = 2$. In particular, the graph of F_w is a parabola whose global minimum occurs at the critical point $c_w = 2(1-w)$. The value of F_w at that critical point is given by

$$y_w := F_w(c_w) = 4w(1-w).$$

Thus we have found all optimal solutions to (10). We can verify our work using MATLAB which gives us the following table and plot:

w	exit flag	x	f1	f2
0	1	2	4	2.8847e-16
0.1	1	1.8	3.24	0.04
0.2	1	1.6	2.56	0.16
0.3	1	1.4	1.96	0.36
0.4	1	1.2	1.44	0.64
0.5	1	1	1	1
0.6	1	0.8	0.64	1.44
0.7	1	0.6	0.36	1.96
0.8	1	0.4	0.16	2.56
0.9	1	0.2	0.04	3.24
1	1	-7.1473e-09	5.1084e-17	4



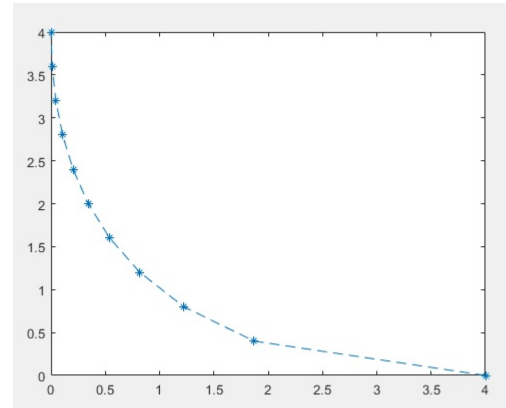
3.2 Epsilon-constraint Method

Next we solve (10) using the epsilon-constraint method. In this case, we only need to consider the following epsilon-constraint problem

$$\begin{array}{ll} \text{minimize} & f_1(x) \\ \text{subject to} & x \in X \\ & f_2(x) - \varepsilon \leq 0 \end{array} \tag{11}$$

where $0 \leq \varepsilon \leq 4$ by the exact same argument as in the previous two cases. . Then f_2 is convex in the new feasible region remains convex, so this just becomes a convex problem. Furthermore one can show that (**??**) has a unique (local and global) optimal solution in this case, and so we can apply part 2 of Proposition (1.2). MATLAB produces the following table and plot:

e	exit flag	x	f1	f2
0	2	2	4	2.279e-16
0.4	1	1.3675	1.8702	0.4
0.8	1	1.1056	1.2223	0.8
1.2	1	0.90455	0.81822	1.2
1.6	1	0.73509	0.54036	1.6
2	1	0.58579	0.34315	2
2.4	1	0.45081	0.20323	2.4
2.8	1	0.32668	0.10672	2.8
3.2	1	0.21115	0.044582	3.2
3.6	1	0.10263	0.010534	3.6
4	1	0.00069373	4.8126e-07	3.9972



4 My BOP

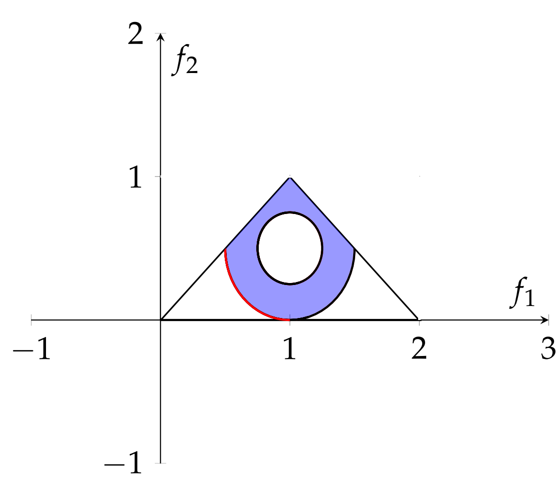
For this problem, we want to create our own BOP which can be used for testing purposes. Let $f_1 = f_1(\boldsymbol{x})$ and $f_2 = f_2(\boldsymbol{x})$ be functions on $\mathbb{R}^2$ which are to be determined, and let

$$\begin{aligned}
e_1 &= f_2 \\
e_2 &= f_2 - f_1 \\
e_3 &= f_1 + f_2 - 2 \\
S &= (f_1 - 1)^2 + (f_2 - 1/2)^2 - 1/4 \\
s &= (f_1 - 1)^2 + (f_2 - 1/2)^2 - 1/16.
\end{aligned}$$

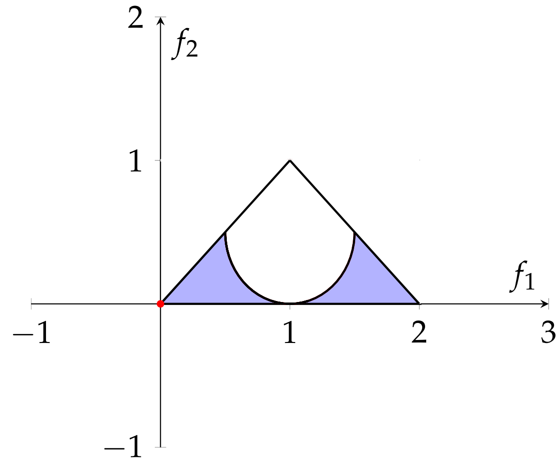
Consider the following BOP:

$$\begin{aligned}
\text{minimize} \quad & [f_1(\boldsymbol{x}), f_2(\boldsymbol{x})] \\
\text{subject to} \quad & e_1 \geq 0 \\
& e_2 \leq 0 \\
& e_3 \leq 0 \\
& S \leq 0 \\
& s \geq 0 \\
& 0 \leqq \boldsymbol{x} \leqq 2
\end{aligned} \tag{12}$$

Thus the feasible region for this BOP is just $X = [0,2]^2$. We draw the outcome space $Y = \boldsymbol{f}(X)$ below shaded in blue together with the Pareto front Y_N shaded in red:



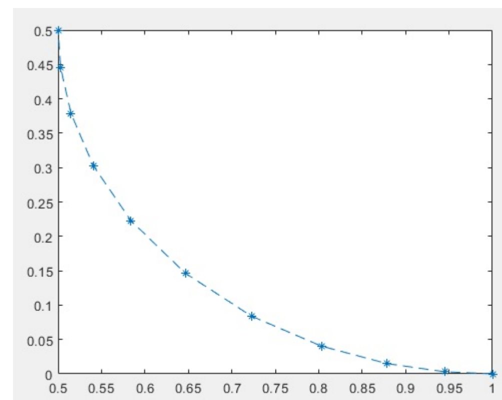
One reason why we think this BOP is interesting and could serve as a good test problem is because of the freedom we have in choosing the outcome set Y together with the Pareto front Y_N. For example, by changing $S \leq 0$ to $s \geq 0$ in (12), then instead we get this for Y and Y_N:



There are many other choices we can make to get other interesting regions like this.

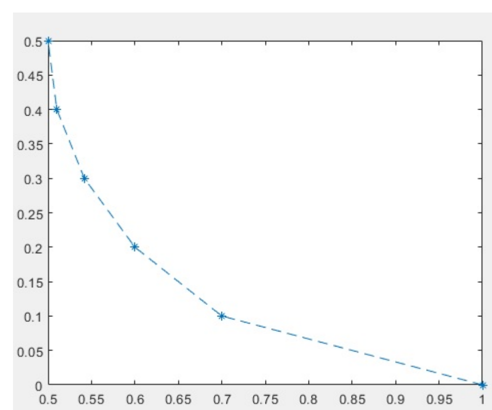
Finally, let $f_1 = x_1$ and $f_2 = x_2$ and let's verify in MATLAB that the red segment which we drew above is indeed the Pareto front for (12). First let's do this using the weighted-sum method:

w	exit flag	x1	x2	f1	f2
0	1	1	6.0045e-08	1	6.0045e-08
0.1	1	0.94478	0.0030586	0.94478	0.0030586
0.2	1	0.87872	0.014931	0.87872	0.014931
0.3	1	0.80304	0.04043	0.80304	0.04043
0.4	1	0.72265	0.083975	0.72265	0.083975
0.5	1	0.64645	0.14645	0.64645	0.14645
0.6	1	0.58397	0.22265	0.58397	0.22265
0.7	1	0.54043	0.30304	0.54043	0.30304
0.8	1	0.51493	0.37873	0.51493	0.37873
0.9	1	0.50306	0.44478	0.50306	0.44478
1	1	0.5	0.49951	0.5	0.49951



Next let's do this using the epsilon-constraint method:

e	exit flag	x1	x2	f1	f2
0	2	1	3.7908e-13	1	3.7908e-13
0.1	1	0.7	0.1	0.7	0.1
0.2	1	0.6	0.2	0.6	0.2
0.3	1	0.54174	0.3	0.54174	0.3
0.4	1	0.5101	0.4	0.5101	0.4
0.5	1	0.5	0.49943	0.5	0.49943
0.6	1	0.5	0.49963	0.5	0.49963
0.7	1	0.5	0.49958	0.5	0.49958
0.8	1	0.5	0.49958	0.5	0.49958
0.9	1	0.5	0.49951	0.5	0.49951
1	1	0.5	0.49951	0.5	0.49951



Concluding Remarks

We end with some concluding remarks. Even though we've found more success calculating Pareto fronts using the epsilon-constraint method over the weighted-sum method, we note that BOP2 and BOP4 points to one advantage that the weighted-sum method has over the epsilon-constraint: one can often choose the weighted vector $\boldsymbol{w}$ to be uniformally distributed in $[0, 1]$ (i.e. the w_i partition $[0, 1]$ into equal parts) in the weighted-sum method, however there is no obvious way of choosing $\boldsymbol{\varepsilon}$ like this in the epsilon-constraint method. Indeed, one often has to work out what the right choice is for a particular problem (for instance, choosing $\varepsilon \geq 0$ would not have worked in the epsilon-constraint method BOP2, we really did need to pick $\varepsilon \leq -100$.

5 Appendix

We coded a lot in this project, so instead of overfilling this Appendix with tons of code, we decided to post all of our MATLAB code at our github page where the reader can view or download the code for themselves via this URL:

https://github.com/mnelso32/Expository-Notes/tree/main/Mathematics/MATLAB

References

[DPM02] Kalyanmoy Deb, Amrit Pratap, and T. Meyarivan. "Constrained Test Problems for Multi-Objective Evolutionary Optimization". Page 4.

[DPM02-1] Kalyanmoy Deb, Amrit Pratap, and T. Meyarivan. "Constrained Test Problems for Multi-Objective Evolutionary Optimization". Page 3.

[HHBW] Simon Huband, Philip Hingston, Luigi Barone, Lyndon While. A Review of Multi-Objective Test Problems and a Scalable Test Problem Toolkit". Page 491.