

Exercise 9 – Solutions

30.11.2021

#1 Finding Pareto-optimal solutions with Excel

a) The MOO problem is

$$\begin{aligned} & \text{v-max}_{[x_1 \ x_2] \in \mathbb{N}^2} \begin{bmatrix} 2x_1 + x_2 \\ x_2 \end{bmatrix} \\ & \text{s.t.} \\ & 2x_1 + 3x_2 \leq 20 \end{aligned}$$

b) See the Excel model solution on MyCourses site.

c) The overall value of an alternative (ticket buying strategy) is given by the function $V(y_1, y_2) = w_1 v_1^N(y_1) + w_2 v_2^N(y_2)$, where $v_1^N(y_1) = \frac{y_1}{20}$ and $v_2^N(y_2) = \frac{y_2}{6}$. The given preference statement implies $V(y_1 + 1, y_2) - V(y_1, y_2) = V(y_1, y_2 + 2) - V(y_1, y_2)$ from which it can be solved $\frac{w_1}{20} = \frac{2w_2}{6} \Rightarrow w_1 \approx 0.87, w_2 \approx 0.13$. With these weights, the best ticket buying strategy is $(y_1, y_2) = (10, 0)$ as it provides the largest overall value among the relevant decision alternatives (the PO solutions), see the Excel model solution for concrete comparisons.

#2 Finding Pareto-optimal solutions with MATLAB

a)

The problem can be formulated as a 2-objective integer linear programming model:

$$\begin{aligned} & \text{v-max}_{x \in \mathbb{N}^m} \begin{bmatrix} \sum_{j=1}^m \mu_j x_j \\ \sum_{j=1}^m -\sigma_j^2 x_j \end{bmatrix} \\ & \text{s.t.} \\ & \sum_{j=1}^m c_j x_j \leq b \\ & \sum_{j=1}^m (-r_{kj} c_j) x_j \leq -0.2b \quad \forall k = 1, 2, 3 \\ & 0 \leq x_j \leq 1 \quad \forall j = 1, \dots, m \end{aligned}$$

Using the matrix form

$$\text{v-max}_{x \in \mathbb{N}^m} \{Cx \mid Ax \leq B\}, C \in \mathbb{R}^{n \times m}, A \in \mathbb{R}^{q \times m}, B \in \mathbb{R}^q$$

we get

$$C = \begin{bmatrix} \mu_1 & \dots & \mu_m \\ -\sigma_1^2 & \dots & -\sigma_m^2 \end{bmatrix} \in \mathbb{R}^{2 \times m}$$

$$A = \begin{bmatrix} c_1 & \dots & c_m \\ -r_{11}c_1 & \dots & -r_{1m}c_m \\ -r_{21}c_1 & \dots & -r_{2m}c_m \\ -r_{31}c_1 & \dots & -r_{3m}c_m \\ I_{m \times m} \\ -I_{m \times m} \end{bmatrix} \in \mathbb{R}^{(2m+4) \times m}$$

$$B = \begin{bmatrix} b \\ -0.2b \\ -0.2b \\ -0.2b \\ 1_{m \times 1} \\ 0_{m \times 1} \end{bmatrix} \in \mathbb{R}^{2m+4}$$

where I is the identity matrix, 1 a vector of ones, 0 a vector of zeros.

b)

i) Weighted sum –algorithm (see also slide 16 of Lecture 8)

1. Set $X = \emptyset$
2. Take random points $\tilde{\lambda}_1, \tilde{\lambda}_2 \sim U(0, 1)$, and define $\lambda_1 = \frac{\tilde{\lambda}_1}{\tilde{\lambda}_1 + \tilde{\lambda}_2}, \lambda_2 = \frac{\tilde{\lambda}_2}{\tilde{\lambda}_1 + \tilde{\lambda}_2}$.
3. Solve

$$\max_{x \in \mathbb{N}^m} \{\tilde{c}x | Ax \leq B\},$$

where $\tilde{c} = [\lambda_1 \ \lambda_2]C$ and C, A and B are defined above.

4. $X = X \cup \{x^*\}$, where x^* is the optimum point obtained in step 3.
5. Return to step 2.

ii) Weighted max-norm –algorithm (see also slides 20-21 of Lecture 8)

1. Set $X = \emptyset$
2. Solve utopian vector $f^* = (\mu^*, \sigma^{2*})$ (two ILP problems):

(I)

$$\mu^* = \max_{x \in \mathbb{N}^m} \{[\mu_1 \ \dots \ \mu_m]x | Ax \leq B\} + \varepsilon, \quad \varepsilon > 0,$$

(II)

$$\sigma^{2*} = \max_{x \in \mathbb{N}^m} \{[-\sigma_1^2 \dots -\sigma_m^2]x | Ax \leq B\} - \varepsilon, \quad \varepsilon > 0,$$

where A and B are defined above.

3. Take random points $\tilde{\lambda}_1, \tilde{\lambda}_2 \sim U(0, 1)$, and define $\lambda_1 = \frac{\tilde{\lambda}_1}{\tilde{\lambda}_1 + \tilde{\lambda}_2}, \lambda_2 = \frac{\tilde{\lambda}_2}{\tilde{\lambda}_1 + \tilde{\lambda}_2}$.
4. Solve MILP:

$$\max_{x \in \mathbb{N}^m, \Delta \in \mathbb{R}} \left\{ \begin{bmatrix} 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ \Delta \end{bmatrix} \mid \tilde{A} \begin{bmatrix} x \\ \Delta \end{bmatrix} \leq [-\lambda_1 \mu^* - \lambda_2 \sigma^{2*} \ B]^T \right\},$$

where

$$\tilde{A} = \begin{bmatrix} -\lambda_1 [\mu_1 \dots \mu_m] & -1 \\ \lambda_2 [\sigma_1^2 \dots \sigma_m^2] & -1 \\ A & 0_{(2m+4) \times 1} \end{bmatrix},$$

and A and B are defined above.

5. $X = X \cup \{x^*\}$, where (x^*, Δ^*) is the optimum point obtained in step 4.
6. Return to step 3.

Note that the MILP solved in step 4 is equivalent to the problem

(*)

$$\max_{x \in \mathbb{N}^m, \Delta \in \mathbb{R}} -\Delta$$

s.t.

$$\lambda_1(\mu^* - f_1(x)) \leq \Delta,$$

$$\lambda_2(\sigma^{2*} - f_2(x)) \leq \Delta,$$

$$Ax \leq B,$$

where $f_1(x) = [\mu_1 \dots \mu_m]x$ and $f_2(x) = [-\sigma_1^2 \dots -\sigma_m^2]x$.

On the other hand, Problem (*) is equivalent to the following formulation (see slide 20 of Lecture 8):

(**)

$$\min_{x \in X} \|f^* - f(x)\|_{\max}^\lambda = \min_{x \in X} \max \{ \lambda_1(\mu^* - f_1(x)), \lambda_2(\sigma^{2*} - f_2(x)) \}, X = \{x \in \mathbb{N}^m | Ax \leq B\}$$

But why are the problems (*) and (**) equivalent?

One way to understand it is this: Let x^* be the optimal solution to (**). That is,

$$\|f^* - f(x^*)\|_{\max}^\lambda = \min_{x \in X} \|f^* - f(x)\|_{\max}^\lambda = \Delta^*.$$

Then, it applies that $\Delta^* \geq \lambda_1(\mu^* - f_1(x^*))$ and $\Delta^* \geq \lambda_2(\sigma^{2*} - f_2(x^*))$. So, (x^*, Δ^*) is a feasible solution to (*) as well. Suppose that (x', Δ') is the optimal solution to (*). Then,

$$\Delta' \geq \|f^* - f(x')\|_{\max}^\lambda \geq \min_{x \in X} \|f^* - f(x)\|_{\max}^\lambda = \Delta^*.$$

If $\Delta' > \Delta^*$, there would be a conflict, because then (x', Δ') would be worse solution to (*) than (x^*, Δ^*) . Thus, it must be so that $\Delta' = \Delta^*$. Hence, the optimal solutions to (*) and (**) are the same.

c)

Note that `intlinprog` allows one to define the lower and upper bounds of the decision variables explicitly with the input variables “LB” and “UB”. Thus, one does not need to define the constraint $0 \leq x_i \leq 1$ with the help of the identity matrices $I_{m \times m}$ and $-I_{m \times m}$ as was now done in part a).

1000 repetitions with the both algorithms produce in total 77 Pareto optimal solutions that differ from each other. The projects that are included in all found Pareto-optimal solutions are 1, 2, 23, and 39. Note that these results are only examples of the results that one may obtain. The random nature of the parameters and on each iteration step changes the results on different runs.

