Deterministic OR Models

CO 370

Bertrand Guenin

LaTeXed by *Sibelius Peng*

Preface

Disclaimer Much of the information on this set of notes is transcribed directly/indirectly from the lectures of CO 370 during Fall 2021 as well as other related resources. I do not make any warranties about the completeness, reliability and accuracy of this set of notes. Use at your own risk.

What is operations research (OR)? There's no standard definitions for it. One particular definition: use of mathematical models to make complex decisions for real life problems. The origin is British military in WW2. OR is actually everywhere today. Key milestone: Simplex algorithm (1947).

Recall optimization problem is of the form:

$$\begin{aligned} \max \quad & f(x) \\ \text{s.t.} \quad & \text{a set of constraints} \end{aligned}$$

There are some applications: mail delivery, machine scheduling, inventory problem, network design, facility location, class scheduling, portfolio optimization, surgery planning, sensor location.

For any questions, send me an email via https://notes.sibeliusp.com/contact.

You can find my notes for other courses on https://notes.sibeliusp.com/.

Sibelius Peng

Contents

Part I:

Formulations

1

LP formulations

1.1 Production problem

Products $J = \{1, \ldots, n\}$
Resources $I = \{1, \ldots, m\}$

Data:

- $\forall j \in J : c_j =$ value of unit of product j
- $\forall i \in I : b_i =$ number of units of resource i available
- $\forall i \in I, \forall j \in J : a_{ij} =$ number of units of resource i going to product j

Goal: maximize values of product made subject to available resources

Var: $x_j =$ number of units of product j produced

Then problem is

$$\begin{array}{lll} \max & \sum_{j \in J} c_j x_j & \\ \text{s.t.} & \sum_{j \in J} a_{ij} x_j \leq b_i & (i \in I) \\ & x_j \geq 0 & (j \in J) \end{array}$$

Now let's generalize this problem to have more than one period.

Products $J = \{1, \ldots, n\}$
Resources $I = \{1, \ldots, m\}$
Periods $K = \{1, \ldots, p\}$

Then we have **data**

- $\forall j \in J, k \in K$: $c_{jk} =$ unit value of product j in period k
- $\forall i \in I, k \in K$: $w_{ik} =$ unit price for resource i in period k
- $\forall i \in I, j \in J$: $a_{ij} =$ number of units of resource i going into product j

and the **goal**: decide how much of each resource to buy & how much of each product to make during each period, to maximize total profit. Unused resources are available at next time period.

Var:

- p_{ik} = number of units of resource i, purchased at start of period k
- x_{jk} = number of units of product j made in period k
- z_{ik} = number of units of resource i at the end of period k

$$\text{Profit} = \sum_{k\in K}\left[\sum_{j\in J} c_{jk}x_{jk} - \sum_{i\in I} w_{ik}p_{ik}\right] \tag{1.1}$$

Then we keep track of resources: for $i \in I$, $k \in K$

$$z_{ik} = z_{i(k-1)} + p_{ik} - \sum_{j\in J} a_{ij}x_{jk} \tag{1.2}$$

and we define for $i \in I$,

$$z_{i0} = 0 \tag{1.3}$$

Thus the optimization problem is

$$\begin{aligned} \max \quad & (1.1) \\ \text{s.t.} \quad & (1.2), (1.3) \\ & p, x, z \geq 0 \end{aligned} \tag{P}$$

Remark:
If (P) has a feasible solution of value that is bigger than 0, then (P) is unbounded. So we are missing some assumptions, maybe? For example, b_{ik} = amount of resource i that can be bound during period k. Then we can add constraints: $p_{ik} \leq b_{ik}$.

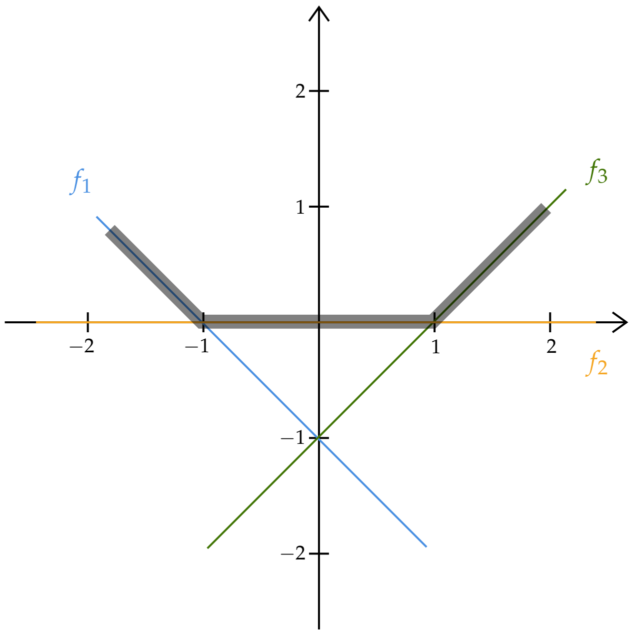
1.2 Minimax

Consider the problem of the form

$$\begin{aligned} & \min_x \max\{f_1(x), \ldots, f_k(x)\} := g(x) \\ & \text{s.t.} \ \cdots \end{aligned}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$.

Example:
$f_1(x) = -x - 1, f_2(x) = 0, f_3(x) = x - 1$. Then $\max\{f_1(x), f_2(x), f_3(x)\}$ is as follows



A motivation

- $\forall i \in [k]$, $f_i(x)$ = completion time for task i.
- Project consists of task $1, \ldots, k$.
- $g(x)$ = completion time of entire project

Note that minimax is not an optimization problem as we defined it. We can revise it as follows

$$\begin{array}{rl} \min & y \\ \text{s.t.} & y \geq f_1(x) \\ & y \geq f_2(x) \\ & \vdots \\ & y \geq f_k(x) \\ & \cdots \end{array}$$

An application minimize a piece-wise linear convex function using linear programming.