

Numerical Optimization, 2020 Fall

Homework 5

Due on 14:59 OCT 27, 2020

请尽量使用提供的 tex 模板, 单纯形法的表格可手绘拍照加入文档.

Production Planning by a Computer Manufacturer

(建议阅读 Bertsimas 教材 “Introduction to Linear Optimization” 的 1.2 节和 5.1 节对应内容)

线性规划建模和求解

公司 Digital Equipment Corporation (DEC) 可以生产 5 种不同的产品 (GP-1, GP-2, GP-3, WS-1, WS-2)。五种产品的生产分别需要两种原件 (disk drives 和 256K boards) 的数量, 以及五种产品的售价如下表:

System	Price	# disk drives	# 256K boards
GP-1	\$60,000	0.3	4
GP-2	\$40,000	1.7	2
GP-3	\$30,000	0	2
WS-1	\$30,000	1.4	2
WS-2	\$15,000	0	1

在实际生产加工中还有以下约束:

1. 五种产品的生产总数不超过 7000;
2. disk drives 原材料的供应量在 3000 个到 7000 个之间;
3. 256K boards 原材料的供应量在 8000 个到 16000 个之间;
4. GP-1 的最大需求不超过 1800 个, GP-3 最大需求不超过 300 个, GP-1,2,3 的总需求不超过 3800 个, WS-1,2 的最大总需求不超过 3200 个; GP-2 的最小需求不低于 500 个, WS-1 的最小需求不低于 500 个, WS-2 的最小需求不低于 400 个。

由于原材料 disk drives 和 256K boards 的总供给量限制, DEC 公司给出了对应的解决方案:

- 对于 disk drives 的供给不足提出了 constrained mode: 仅 GP-2 需要一个 disk drive, WS-1 需要一个 disk drive, 其他产品的生产均不需要 disk drive;

- 对于 256K boards 的供给不足提出了 alternative mode: GP-1 的生产可以用 2 块 alternative boards 来替换 4 块 256K boards, alternative boards 的供给量为 4000 块。其他产品不能使用 alternative boards。

因此, 基于 constrained mode 和 alternative mode, 我们共有四种可选择的生产方案: (方案一):alternative mode & constrained mode, (方案二):alternative mode & unconstrained mode, (方案三): not use alternative mode & constrained mode, (方案四):not use alternative mode & unconstrained mode。

注: 为表述方便, 数量和价格均以“千”为单位。设变量 x_1, \dots, x_5 表示五种产品的生产数量 (千个), 则 $1000x_i$ 应为整数。这里我们忽略整数约束, 因为近似地可以截断解的小数点后三位, 带来的误差忽略不计。

问题一:

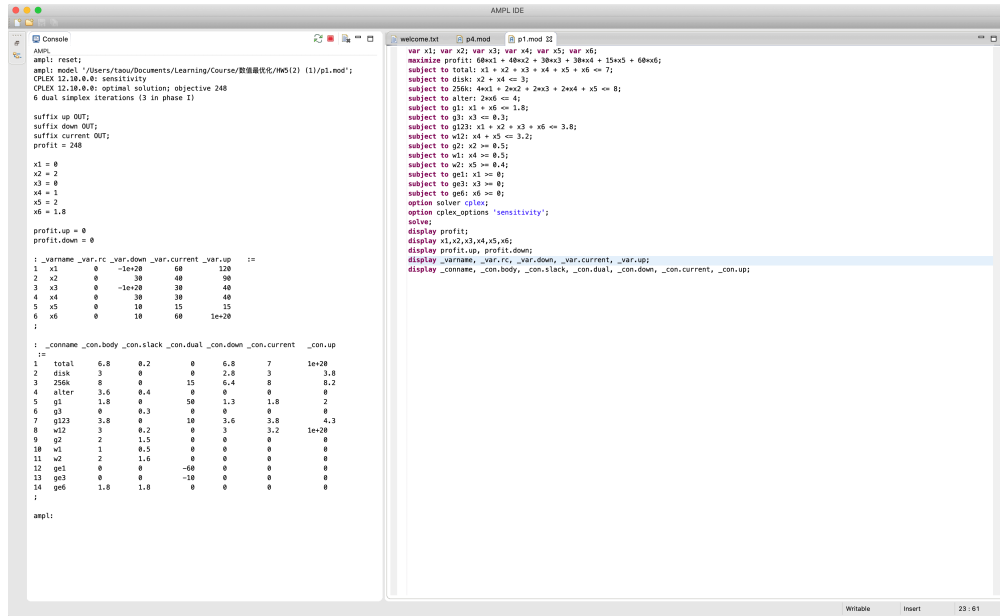
- 若 DEC 公司使用方案一, 写出在满足约束下最大化收益的线性规划问题。(该模型中公司以保守起见, 即, 假设 disk drive 的供给量为 3000 个, 256K boards 的供给量为 8000 个。) [20pts]
- 用 AMPL (CPLEX solver) 求解上述线性规划问题, 给出问题最优解及相应目标函数值。(注: 将程序代码及运行结果截图附在下方) [20pts]

解

- 令 $x = (x_1, x_2, x_3, x_4, x_5, x_6)$, 其中 (x_2, x_3, x_4, x_5) 分别对应 (G2,G3,W1,W2) 的生产数量, (x_1, x_6) 分别对应用 256K board 和 alternative board 生产的 G1 的数量, 依据题目提供的条件, 我们写出对应的线性规划模型

$$\begin{aligned}
 \max_{x \in \mathbb{R}^6} \quad & 60x_1 + 40x_2 + 30x_3 + 30x_4 + 15x_5 + 60x_6 \\
 \text{s.t.} \quad & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 7 \\
 & x_2 + x_4 \leq 3 \\
 & 4x_1 + 2x_2 + 2x_3 + 2x_4 + x_5 \leq 8 \\
 & 2x_6 \leq 4 \\
 & x_1 + x_6 \leq 1.8 \\
 & x_3 \leq 0.3 \\
 & x_1 + x_2 + x_3 + x_6 \leq 3.8 \\
 & x_4 + x_5 \leq 3.2 \\
 & x_2, x_4 \geq 0.5, x_5 \geq 0.4, x_1, x_3, x_6 \geq 0
 \end{aligned} \tag{1}$$

- 代码如下所示, 最优解为 $x = (0, 2, 0, 1, 2, 1.8)$, 最优值为 248k, 即生产 GP-1, GP-2, GP-3, WS-1, WS-2 分别为 (1.8, 2, 0, 1, 2, 1.8), 最大收益为 248k。



灵敏度分析

DEC 公司为了从四种方案中做出选择, 分别求解了四种方案下对应问题的最优解:

Alt. boards	Mode	Revenue	x_1	x_2	x_3	x_4	x_5
no	constr.	145	0	2.5	0	0.5	2
yes	constr.	248	1.8	2	0	1	2
no	unconstr.	133	0.272	1.304	0.3	0.5	2.7
yes	unconstr.	213	1.8	1.035	0.3	0.5	2.7

Table 5.1: Optimal solutions to the four variants of the production planning problem. Revenue is in millions of dollars and the quantities x_i are in thousands.

上述表格中易见, alternative mode 会带来明显收益, 公司应选择该模式. 而对于是否选择 constrained mode 则没那么显然. 此外我们上述考虑的线性规划对于 disk drives 和 256K boards 的供应量的估计是比较保守的. 因此, 下面我们考虑在问题一解的基础上, 增加 disk drives 和 256K boards 的供应量的灵敏度分析问题.

问题二:

- 用线上的单纯形表法求解器求解问题一中线性规划问题, 附上第一张和最后一张单纯形表的截图. [20pts]
(可以选择以下网站:https://www.mathstools.com/section/main/simplex_online_calculator
或 <http://simplex.tode.cz/en/> (需要 vpn))
- 根据上一问中的单纯形表, 分析当 disk drives 和 256K boards 数量的取值在什么范围内, 当前问题的解仍为最优解. 并分析对应的目标函数值将如何变化. [20pts]

(iii) 用 AMPL (CPLEX solver) 做灵敏度分析检验上一问的结论 (disk drives 和 256K boards 数量的取值范围), 给出程序执行结果截图. [20pts]
 (Hint: 查看语句 “option cplex_options ‘sensitivity’;”)

解

(i) 表格如下

Mathstools Widgets

Waiting for next Iteration

Xb0	Cb1	Basis	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16	x17	x18	x19
7	0	x6	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	x7	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
8	0	x8	4	2	2	2	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
4	0	x9	0	0	0	0	0	2	0	0	0	1	0	0	0	0	0	0	0	0	0	0
9...	0	x...	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
3...	0	x...	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1...	0	x...	1	1	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1...	0	x...	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
1...	-1	x...	0	1	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	1	0	0
1...	-1	x...	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	1	0
2...	-1	x...	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	1
		Z...	0	-1	0	-1	-1	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0

Next Step >

Go to End >>

View Android Version

Status: Waiting for next Iteration

Pivot 1

Step 1

Optimal 0

Elapsed 0.0 segs.

Solution x6=7, x7=3, x8=8, x9=4, x10=9/5, x11=3/10, x12=19/5, x13=16/5, x17=1/2, x18=1/2, x19=2/5

Mathstools Widgets

Infinite optimal finite solution found

Xb0	Cb1	Basis	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16
1/5	0	x6	-4	0	-1	0	0	0	1	1	-1	0	-1	0	0	0	0	0	0
1/2	0	x15	0	0	-1	0	0	0	0	1	0	0	1	0	-1	0	0	1	0
8/5	0	x16	4	0	2	0	0	0	0	-2	1	0	0	0	0	0	0	0	1
2/5	0	x9	-2	0	0	0	0	0	0	0	0	1	-2	0	0	0	0	0	0
9/5	60	x5	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0
3/...	0	x11	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0
3/2	0	x14	0	0	1	0	0	0	0	0	0	0	-1	0	1	0	1	0	0
1/5	0	x13	-4	0	-1	0	0	0	0	1	-1	0	-1	0	1	1	0	0	0
2	40	x1	0	1	1	0	0	0	0	0	0	0	-1	0	1	0	0	0	0
1	30	x3	0	0	-1	1	0	0	0	1	0	0	1	0	-1	0	0	0	0
2	15	x4	4	0	2	0	1	0	0	-2	1	0	0	0	0	0	0	0	0
		zj...	60	0	10	0	0	0	0	15	0	50	0	0	10	0	0	0	0

View Android Version

Status: Infinite optimal finite solution found

Pivot 1
Step 12
Optimal 247.99999999999997
Elapsed 0.02 segs.
Solution x6=1/5, x15=1/2, x16=8/5, x9=2/5, x5=9/5, x11=3/10, x14=3/2, x13=1/5, x1=2, x3=1, x4=2

(ii) 根据最后的单纯形表可知，基变量为 $x_6, x_{15}, x_{16}, x_9, x_5, x_{11}, x_{14}, x_{13}, x_1, x_3, x_4$ ，对应的基矩阵及其逆为

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 1 \\ 0 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B^{-1} = \begin{bmatrix} 1 & 1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & -1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

对于 disk drive 和 256k board, 对应着 B^{-1} 的第二列和第三列, 根据可行性条件

$$B^{-1}(b + \delta) \geq 0$$

可得

$$x_{B(2)} + \delta\beta_{2i} \geq 0$$

$$x_{B(2)} + \delta\beta_{3i} \geq 0$$

等价于

$$\max_{\beta_{2i} > 0} \left(-\frac{x_{B(2)}}{\beta_{2i}} \right) \leq \delta \leq \min_{\beta_{2i} > 0} \left(-\frac{x_{B(2)}}{\beta_{2i}} \right)$$

$$\max_{\beta_{3i} > 0} \left(-\frac{x_{B(3)}}{\beta_{3i}} \right) \leq \delta \leq \min_{\beta_{3i} > 0} \left(-\frac{x_{B(3)}}{\beta_{3i}} \right)$$

由此可得

$$-0.2 \leq \delta_{disk} \leq 0.8, -1.6 \leq \delta_{256k} \leq 0.2 \quad (\text{对应 } \beta_{2i}, \beta_{3i})$$

即 disk 的取值在 (2.8,3.8) 之间, 256k board 取值在 (6.4,8.2) 之间。

对于目标函数的变化, 由公式 $c_B^T B^{-1} e_i$ 可得

$$\text{disk: } c_B^T B^{-1} e_2 = (0, 0, 0, 0, 60, 0, 0, 0, 40, 30, 15) \cdot (1, 1, -2, 0, 0, 0, 0, 1, 0, 1, -2) = 0$$

$$\text{256k board: } c_B^T B^{-1} e_3 = (0, 0, 0, 0, 60, 0, 0, 0, 40, 30, 15) \cdot (-1, 0, 1, 0, 0, 0, 0, -1, 0, 0, 1) = 15$$

即改变单位 (1k) 的 disk 不会影响收益, 改变单位 (1k) 的 256k board 会增加 15k 的收益。

- (iii) 由灵敏度分析结果可知 (line2,3), disk 的取值在 (2.8,3.8) 之间, 256k board 取值在 (6.4,8.2) 之间, 验证了第二问中的结论。

```

ampl: model '/Users/taou/Documents/Learning/Course/数值最优化/Hw5(2) (1)/p1.mod';
CPLEX 12.10.0.0: sensitivity
CPLEX 12.10.0.0: optimal solution; objective 248
6 dual simplex iterations (3 in phase I)

suffix up OUT;
suffix down OUT;
suffix current OUT;
profit = 248

x1 = 0
x2 = 2
x3 = 0
x4 = 1
x5 = 2
x6 = 1.8

profit.up = 0
profit.down = 0

: _varname _var.rc _var.down _var.current _var.up :=
1 x1 0 -1e+20 60 120
2 x2 0 30 40 90
3 x3 0 -1e+20 30 40
4 x4 0 30 30 40
5 x5 0 10 15 15
6 x6 0 10 60 1e+20
;

: _conname _con.body _con.slack _con.dual _con.down _con.current _con.up :=
1 total 6.8 0.2 0 6.8 7 1e+20
2 disk 3 0 0 2.8 3 3.8
3 256k 8 0 15 6.4 8 8.2
4 alter 3.6 0.4 0 0 0 0
5 g1 1.8 0 50 1.3 1.8 2
6 g3 0 0.3 0 0 0 0
7 g123 3.8 0 10 3.6 3.8 4.3
8 w12 3 0.2 0 3 3.2 1e+20
9 g2 2 1.5 0 0 0 0
10 w1 1 0.5 0 0 0 0
11 w2 2 1.6 0 0 0 0
12 ge1 0 0 -60 0 0 0
13 ge3 0 0 -10 0 0 0
14 ge6 1.8 1.8 0 0 0 0
;

ampl:

```