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 Corperation (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 $x_1, \cdots, x_5$ 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:

(i) 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]

(ii) 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]

(i)

$$
\begin{aligned}
\max_{x_1,x_2,x_3,x_4,x_5} \quad & 60x_1 + 40x_2 + 30x_3 + 30x_4 + 15x_5 & & \\
s.t. \quad & x_1 + x_2 + x_3 + x_4 + x_5 & \leq 7 \quad & \text{(CPU availability)} \\
& 2x_1 & \leq 4 \quad & \text{(alternative boards availability)} \\
& 2x_2 + 2x_3 + 2x_4 + x_5 & \leq 8 \quad & \text{(256K boards availability)} \\
& x_2 + x_4 & \leq 3 \quad & \text{(disk drives availability)} \\
& x_1 & \leq 1.8 \quad & \text{(max demand for GP-1)} \\
& x_3 & \leq 0.3 \quad & \text{(max demand for GP-3)} \\
& x_1 + x_2 + x_3 & \leq 3.8 \quad & \text{(max demand for GP)} \\
& x_4 + x_5 & \leq 3.2 \quad & \text{(max demand for WS)} \\
& x_2 & \geq 0.5 \quad & \text{(min demand for GP-2)} \\
& x_4 & \geq 0.5 \quad & \text{(min demand for WS-1)} \\
& x_5 & \geq 0.4 \quad & \text{(min demand for WS-2)} \\
& x_1, x_2, x_3, x_4, x_5 & \geq 0 \quad &
\end{aligned}
$$

(ii)

```
var X1;
var X2;
var X3;
var X4;
var X5;

maximize Profit: 60 * X1 + 40 *X2 + 30 * X3 + 30 * X4 + 15 * X5;
subject to CPU: X1 + X2 + X3 + X4 + X5 <= 7;
subject to alter: 2 * X1 <= 4;
subject to 256K: 2 * X2 + 2 * X3 + 2 * X4 + X5 <= 8;
subject to disk: X2 + X4 <= 3;
subject to max_GP1: X1 <= 1.8;
subject to max_GP3: X3 <= 0.3;
subject to max_GP: X1 + X2 + X3 <= 3.8;
subject to max_WS: X4 + X5 <= 3.2;
subject to min_GP2: X2 >= 0.5;
subject to min_WS1: X4 >= 0.5;
subject to min_WS2: X5 >= 0.4;
subject to min_GP1: X1 >= 0;
subject to min_GP3: X3 >= 0;
```

```
ampl: model test/prod0.mod;
ampl: option solver '/home/fan/Downloads/ampl.linux64/cplex';
ampl: solve;
CPLEX 12.10.0.0: optimal solution; objective 248
4 dual simplex iterations (2 in phase I)
ampl: display X1, X2, X3, X4, X5;
X1 = 1.8
X2 = 2
X3 = 0
X4 = 1
X5 = 2
```

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:

(i) 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))

(ii) 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’;”)

(i)
完整的计算流程见http://simplex.tode.cz/en/408if9knef5.

#3 FACTORY RESTRICTIONS EQUATIONS

Maximize $60x_1 + 40x_2 + 30x_3 + 30x_4 + 15x_5$

subject to

$$
\begin{aligned}
x_1 + x_2 + x_3 + x_4 + x_5 + x_6 &= 7\\
2x_1 + x_7 &= 4\\
2x_2 + 2x_3 + 2x_4 + x_5 + x_8 &= 8\\
x_2 + x_4 + x_9 &= 3\\
x_{10} + x_3 &= 3/10\\
x_1 + x_{11} + x_2 + x_3 &= 19/5\\
x_{12} + x_4 + x_5 &= 16/5\\
-x_{13} + x_2 + y_1 &= 1/2\\
-x_{14} + x_4 + y_2 &= 1/2\\
-x_{15} + x_5 + y_3 &= 2/5\\
x_1 + x_{16} &= 9/5
\end{aligned}
$$

#4 SIMPLEX TABLE 1

	x_1	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	y_1	y_2	y_3	b	t
x_{10}	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	3/10	–
x_{11}	1	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	19/5	19/5
x_{12}	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	16/5	–
x_{16}	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	9/5	–
x_6	1	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	7	7
x_7	2	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	4	–
x_8	0	0	0	0	0	0	0	0	2	2	2	1	0	0	1	0	0	0	0	8	4
x_9	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	3	3
y_1	0	0	0	0	-1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1/2	1/2
y_2	0	0	0	0	0	-1	0	0	0	0	1	0	0	0	0	0	0	1	0	1/2	–
y_3	0	0	0	0	0	0	-1	0	0	0	0	1	0	0	0	0	0	0	1	2/5	–
z	-60	0	0	0	0	0	0	0	-40	-30	-30	-15	0	0	0	0	0	0	0	0	
z'	0	0	0	0	1	1	1	0	-1	0	-1	-1	0	0	0	0	0	0	0	-7/5	

#14 SIMPLEX TABLE 11

	x_1	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	b
x_{10}	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	3/10
x_{13}	0	0	1	0	1	0	0	-1	0	1	0	0	0	0	0	0	3/2
x_{12}	0	0	1	1	0	0	0	-1	0	-1	0	0	0	0	-1	1	1/5
x_1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	9/5
x_6	0	0	0	0	0	0	0	-1	0	-1	0	0	1	0	-1	1	1/5
x_7	0	0	0	0	0	0	0	-2	0	0	0	0	0	1	0	0	2/5
x_{15}	0	0	0	0	0	0	1	0	0	2	0	0	0	0	1	-2	8/5
x_{14}	0	0	-1	0	0	1	0	1	0	-1	0	0	0	0	0	1	1/2
x_2	0	0	1	0	0	0	0	-1	1	1	0	0	0	0	0	0	2
x_4	0	0	-1	0	0	0	0	1	0	-1	1	0	0	0	0	1	1
x_5	0	0	0	0	0	0	0	0	0	2	0	1	0	0	1	-2	2
z	0	0	10	0	0	0	0	50	0	10	0	0	0	0	15	0	248

SOLUTION

$\mathbf{x}^{(10)}$ = (9/5 , 3/10 , 0, 1/5 , 3/2 , 1/2 , 8/5 , 0, 2, 0, 1, 2, 1/5 , 2/5 , 0, 0)
$\mathbf{z}^{(10)}$ = 248

OPTIMAL SOLUTION

$\mathbf{x}^*$ = (9/5 , 3/10 , 0, 1/5 , 3/2 , 1/2 , 8/5 , 0, 2, 0, 1, 2, 1/5 , 2/5 , 0, 0)
$\mathbf{z}^*$ = 248

(ii) 由第一张单纯形表可知:256K boards 约束对应的对偶变量和 B^{-1} 的那一列为单纯形表中 x_8 对应的那一列, 记其下标为 i; 相应的 disk drives 约束对应的对偶变量和 B^{-1} 的那一列为单纯形表中 x_9 对应的那一列, 记其下标为 j. 令 b_i 表示 B^{-1} 矩阵的第 i 列, δ_i, δ_j 分别为两个约束右端项的变化量.

由最后一张单纯形表可知:

$$\begin{aligned} x_B^* &= (x_{10}^*, x_{13}^*, x_{12}^*, x_1^*, x_6^*, x_7^*, x_{15}^*, x_{14}^*, x_2^*, x_4^*, x_5^*) \\ &= (0.3, 1.5, 0.2, 1.8, 0.2, 0.4, 1.6, 0.5, 2, 1, 2) \\ b_i &= (0, 0, -1, 0, -1, 0, 1, 0, 0, 0, 1) \\ b_j &= (0, 0, 1, 0, 1, 0, -2, 1, 0, 1, -2) \\ \lambda_i &= 15, \ \ \lambda_j = 0. \end{aligned}$$

则

$$\begin{aligned} -1.6 = \max_{\{k|[b_i]_k>0\}} (-\frac{x_{B(k)}}{[b_i]_k}) &\le \delta_i \le \min_{\{k|[b_i]_k<0\}} (-\frac{x_{B(k)}}{[b_i]_k}) = 0.2 \\ -0.2 = \max_{\{k|[b_j]_k>0\}} (-\frac{x_{B(k)}}{[b_j]_k}) &\le \delta_j \le \min_{\{k|[b_j]_k<0\}} (-\frac{x_{B(k)}}{[b_j]_k}) = 0.8. \end{aligned}$$

即, # 256 boards $\in [6.4, 8]$, # disk drives $\in [2.8, 3.8]$.

在这组最优解下, 改变 disk drives 的供给不能带来更大收益, 而将 256 boards 供给提高 0.2, 可以多带来 $0.2 * 15 = 3$ million 的收益.

(iii)

```
ampl: model test/prod0.mod;
ampl: option solver '/home/fan/Downloads/ampl.linux64/cplex';
ampl: option cplex_options 'sensitivity';
ampl: solve;
CPLEX 12.10.0.0: sensitivity
CPLEX 12.10.0.0: optimal solution; objective 248
4 dual simplex iterations (2 in phase I)

suffix up OUT;
suffix down OUT;
suffix current OUT;
ampl: display 256K.up, 256K.down, 256K.current;
256K.up = 8.2
256K.down = 6.4
256K.current = 8

ampl: display disk\.up, disk.down, disk.current;

syntax error
context:  display  >>> disk\ <<< .up, disk.down, disk.current;
ampl: display disk.up, disk.down, disk.current;
disk.up = 3.8
disk.down = 2.8
disk.current = 3
```

$\#$ 256 boards $\in [6.4, 8]$, $\#$ disk drives $\in [2.8, 3.8]$.