数学建模期末考试

学号：2022211570 姓名：项枫

PART Ⅰ 四选三

一、解:

在二人零和博弈中，给定收益矩阵 $A=[a_{ij}]_{m\times n}$，其中 a_{ij} 表示玩家 1 选择策略 i 而玩家 2 选择策略 j 时玩家 1 的收益，推导混合策略均衡等价于一个线性规划问题以及对应对偶问题的意义如下:

首先，定义玩家 1 和玩家 2 的混合策略分别为向量 x 和 y:

$$x=(x_1,x_2,\dots,x_m)^T$$

$$y=(y_1,y_2,\dots,y_n)^T$$

其中 x_i 和 y_j 分别表示玩家 1 选择策略 i 和玩家 2 选择策略 j 的概率，并且满足:

$$\sum_{i=1}^{m} x_i=1, x_i\geq 0,\ \forall i$$

$$\sum_{j=1}^{n} y_j=1, y_j\geq 0,\ \forall j$$

设 v 为博弈的值，即在均衡状态下的期望收益。玩家 1 的目标是最大化最小收益，而玩家 2 的目标是最小化最大损失。因此，玩家 1 的线性规划问题为:

玩家 1 的线性规划问题（原问题）

$$\begin{aligned} &maximize && v \\ &subject\ to && \sum_{j=1}^{n} a_{ij}\, y_j \geq v, \forall i = 1,2,\dots,m \\ &&& \sum_{j=1}^{n} y_j = 1 \\ &&& y_j \geq 0, \forall j = 1,2,\dots,n \end{aligned}$$

玩家 2 的线性规划问题（对偶问题）

$$\begin{aligned} &minimize && u \\ &subject\ to && \sum_{i=1}^{m} a_{ij}\, x_i \leq u, \forall j = 1,2,\dots,n \\ &&& \sum_{i=1}^{m} x_i = 1 \\ &&& x_i \geq 0, \forall i = 1,2,\dots,m \end{aligned}$$

对偶问题的意义在于:

博弈的值： 原问题和对偶问题的最优解均为博弈的值 v。即 $max\, v = min\, u$。

策略的解释： 原问题中的变量 y_j 表示玩家 2 的混合策略概率，对偶问题中的变量 x_i 表示玩家 1 的混合策略概率。

最优性条件： 原问题和对偶问题的最优解满足互补松弛条件，即如果某个策略概率为正，则对应的约束必须严格满足。

(1)

				min	max
-5	2	1	19	-5	
6	4	4	8	4*	4*
5	4	0	-6	-6	

max	6	4*	4*	19
min			4*	

故纳什均衡点为 $a_{2,2}=a_{2,3}=4$。

（2）

				min	max
	1	2	4	1	
	6	4	3	3*	3*
max	6	4*	4		
min		4*			

显然该矩阵对策没有纯纳什均衡，这时可以求解其混合纳什均衡。

求解:

$$\underset{p\in\Delta_1}{max}\underset{q\in\Delta_2}{min} pMq^{\top}$$

等价于

$$max\ v$$

$s.t.$

$$pM \le v1$$

$$p=(p1,\dots,pm)\in\Delta_1$$

$$1=(1,\dots,1)^{\top}$$

求解得:

$$X^*=\left(\frac{1}{3},\frac{2}{3}\right)$$

$$Y^*=\left(0,\frac{1}{3},\frac{2}{3}\right)$$

$$V_G = v_1 = v_2 = \frac{10}{3}$$

二、解:

（1）变量和参数解释:

x_t：时刻t的人口数量;

r：人口净增长率;

x_m：最大人口容量，即环境能够支持的最大人口数量。

$x(0) = x_0$：初始时刻t=0 时的人口数量。

（2）建模思想:

阻滞增长模型它是对指数增长模型的基本假设进行修改后得到的，它考虑到了这一因素，即在分析人口增长到一定数量后增长率下降的主要原因，自然资源、环境条件等因素对人口的增长起着阻滞作用，并且随着人口的增加，阻滞作用会越来越大。

三、解:

（1）该模型国家安全的概念:

甲乙双方的“国家安全”概念均采用保守定义，即在遭到对方“倾泻性”核打击后，保证有足够的核武器被保存下来以给对方致命的还击。

（2）模型参数x_0、y_0、p_1、p_2的实际意义:

x_0、y_0表示甲乙双方对对方施行一次致命性打击所需的核武器数目；p_1、p_2 ($0 \leqslant p_1$，$p_2 < 1$)表示甲乙双方，其一枚核弹头在遭受对方一枚核弹头袭击后有可能被保存下来的概率，这里假定不同核弹头

在遭受对方一枚核弹头袭击后有可能被保存下来的机会是相对独立的。

（3）参数取值的主要影响因素：

双方的国土、一枚核弹爆炸的破坏力，以及各自的防空能力等。

四、

（一）学习体会

通过一学期的学习，我对《数学建模与模拟》课程有了深刻的体会和认识。这门课程不仅拓展了我的数学知识和技能，还培养了我分析和解决实际问题的能力。首先是数学思维能力的提升，能够更好地抽象和简化复杂问题。其次是编程能力的提高，熟练掌握了Python、MATLAB 等工具，并能高效地实现数学模型。最后也让我对数学建模竞赛产生了浓厚的兴趣，下学期会参加国赛或美赛。

（二）课程投入

我认为对这门课程的投入是非常值得的。这不仅是因为它直接提升了我的数学和编程能力，更重要的是，它教会了我一种解决问题的思维方式。数学建模是一种桥梁，它连接了理论和实际，使我能够将学到的知识真正应用于现实中。

（三）对未来的帮助

这门课程对我的未来有很大的帮助。无论是继续深造还是进入职场，数学建模的知识和技能都是非常有价值的。在研究生阶段，我认为我可以利用建模方法顺利完成许多科研任务，同时，数学建模

提高了我的思考能力，这对我的科研有巨大帮助；在职场上，许多行业都需要数学建模的技能。

（四）思考

在建模过程中，如何在简化模型与保持精确度之间找到平衡，是一个常见但复杂的问题。

（五）建议

增加实际案例分析：希望课堂上可以多讲一些与生活实际结合的案例，可以是学校或社会上发生的重大事件，这样会更吸引同学们的注意力。

加强编程训练：编程是实现数学模型的重要工具，建议课堂上或课后习题中可以增加编程训练的比重，提高同学们的编程能力。

鼓励团队合作：团队合作可以激发创造力和提高解决问题的能力，建议在课程中更多地安排团队项目。

（六）致谢

感谢张老师一学期的辛苦付出，每堂课都认真负责讲授知识，最后，祝老师身体健康、生活美满、工作顺利！

PART Ⅱ 排产排程习题

（一）问题描述和数学规划模型

1、问题描述

考虑如下决策问题，某香皂厂在下一周将要生产洗衣皂和洗脸皂两种产品，已知该工厂每小时能生产 200 千克洗衣皂或者 140 千克洗脸皂，每千克洗衣皂能获利 25 元，每千克洗脸皂能获利 30 元。

已知该工厂每周最多能够生产 6000 千克的洗衣皂和 4000 千克的洗脸皂，且工厂每周最多投入 40 小时来生产香皂，请问该工厂每周需要生产多少千克的洗衣皂和洗脸皂从而能让利润最大化?

2、数学规划模型

以上问题，我们可以建立线性规划的数学模型如下:

（1）集合:

产品集合 P

混合集合 D(由生产效率 $rate$，每千克获利 $profit$，最低生产量 $commit$和最大生产量 $market$四个元素组成)

（2）参数:

工厂每周最大工作时间 $avail$

工厂生产产品 $p \in P$的效率 r_p

每千克产品 $p \in P$获利 P_p

产品 $p \in P$每天最低生产量 c_p

产品 $p \in P$每天最大生产量 m_p

（3）决策变量:

工厂一周生产产品$p \in P$的数量 $c_p \leq make_p \leq m_p$

（4）目标函数:

工厂要最大化利润 $max\sum_{p \in P} p_p \cdot make_p$

（5）约束:

工厂每周工作不得超过最大工作时间 $make_p/r_p \leq avail$

（二）使用 MindOpt APL 进行建模和求解

我们可以将 MindOpt APL 建模代码直接输入如下 cell 里运行。请注意内核（kernel）需要选 MAPL。代码如下：

```
clear model; #清除 model，多次 run 的时候使用
option modelname model/soap; #方便与方法 2 的中间文件生成在同一个目录

#---------建模----------------
# soap.mapl

set PROD := { "soap_face", "soap_laundry" }; #产品集合：洗衣皂
（soap_laundry）、洗脸皂（soap_face）
set D := { "rate", "profit", "commit", "market" }; #混合集合：生产效率
（rate）、每千克获利（profit）、
                                                  #最低生产量（commit）和
最大生产量（market）

param data[PROD * D] :=
                | "rate", "profit", "commit", "market"|  #       生产效率
每千克获利 最低生产量 最大生产量
|"soap_laundry" |  200,    25,       0,        6000    |  #洗衣
皂  200     25         0          6000
|"soap_face"    |  140,    30,       0,        4000    |; #洗脸
皂  140     30         0          4000

param avail := 40; #工厂每周最大工作时间

var Make[<p> in PROD] >= data[p, "commit"] <= data[p, "market"]; #决策变
量：工厂一周生产产品 p∈P 的数量

maximize Total_Profit: sum {<p> in PROD } data[p, "profit"] * Make[p];
#目标函数：工厂要最大化利润

subto Time:
  sum {<p> in PROD} 1/data[p, "rate"] * Make[p] <= avail; #约束：工厂每周
工作不得超过最大工作时间

#-----------------------------

print "----------------用 MindOpt 求解--------------";
option solver mindopt;   # 指定求解用 MindOpt 求解器
solve;                 # 求解
display;
```

```
print "----------------结果--------------";
print "最大利润 = ",sum { <p> in PROD } data[p, "profit"] * Make[p];
```

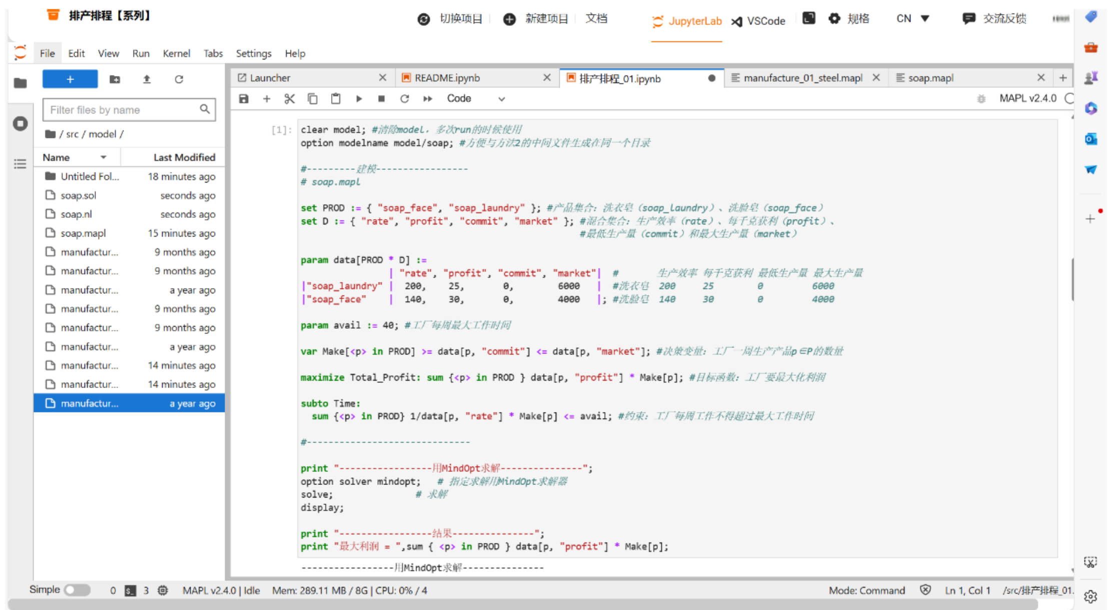

图 1 代码在 MindOpt 中截图

（三）结果解析

从打印的结果，我们可以得到采用如下的生产方案时，利润最大，为 192000 元：

Make@soap_face = 1400.00000

Make@soap_laundry = 6000.000000

```
-----------------用MindOpt求解---------------
Running mindoptampl
wantsol=1
MindOpt Version 1.2.1 (Build date: 20240511)
Copyright (c) 2020-2024 Alibaba Cloud.

Start license validation (current time : 04-JUN-2024 22:14:18).
License validation terminated. Time : 0.003s

Model summary.
 - Num. variables     : 2
 - Num. constraints   : 1
 - Num. nonzeros      : 2
 - Bound range        : [4.0e+01,6.0e+03]
 - Objective range    : [2.5e+01,3.0e+01]
 - Matrix range       : [5.0e-03,7.1e-03]
```

图 2 代码运行结果截图

```
Presolver started.
Presolver terminated. Time : 0.000s

Simplex method started.
Model fingerprint: =E2blNmZul3Z3V2dldnZ

    Iteration       Objective       Dual Inf.      Primal Inf.     Time
            0     2.70000e+05     0.0000e+00      7.4286e-01     0.01s
            1     1.92000e+05     0.0000e+00      0.0000e+00     0.01s
Postsolver started.
Simplex method terminated. Time : 0.002s


OPTIMAL; objective 192000.00
1 simplex iterations

Completed.
Primal Solution:
Make[soap_face] = 1400.00000
Make[soap_laundry] = 6000.00000
-----------------结果---------------
最大利润 = 192000
```

图 3 代码运行结果截图（续）