Universidade do minho

2ºSemestre 2019/20

(MIEI,3ºAno)

Modelos Estocásticos de Investigação Operacional

Trabalho Prático

Identificação do Grupo

Número	Nome completo:	Rubrica:
A84220	Pedro Miguel Queirós Gomes	Pedro Gomes
A78566	Marcos Daniel Teixeira da Silva	Marcos Silva
A64296	André Miguel Vila Cova Ferreira	André Ferreira

Figure 1: Data de entrega : 2020-05-11

Contents

1 Parte 1 do trabalho pratico

1.1 Descrição e formulação do problema

Um empresario que gere duas filiais de um grupo internacional de aluguer de automoveis pretende optimizar o seu modo de gestão visto que ele tem varias decisões que pode tomar, nomeadamente na transferencia de carros entre as suas filiais.Deste modo formulamos um modelo de programção dinamica estocástica para definir a politica otima de transferencia de automoveis diarios entre as filiais que o empresario deve tomar.

Para isso são nos fornecidos varios dados sobre a gestão das filiais tais como, cada carro alugado o empresario é creditado 30 euros, no maximo 3 automoveis podem ser diariamente transferidos entre as filiais e tem um custo de 7 euros/transferencia,as filiais tem um numero maixmo de 12 carros e durante a noite se tiver um numero de automoveis maior do que 8 paga uma taxa fixa de 10 euros devido ao estacionamento.

Visto isto o nosso grupo começou a formular o problema começando por definir os estados e os estagios. Identificamos tambem as 7 decisoes que o empresario poderia efetuar diariamente.

Nós definimos os estados como o numero de veiculos presentes numa dada filial,o que significa que vão de 0 a 12, e os estagios como sendo de 24h, no inicio do dia ou seja de manhã. As 7 decisões que nós identificamos foram não transferir, receber 1 carro, receber 2 carros, receber 3 carros, transferir 1 carro, transferir 2 carros e transferir 3 carros, isto na prespetiva da filial 1 mas que funciona tambem para a filial 2.

De seguida o grupo reparou que devido ao numero de estados elevado e tambem a existirem varias decisões que o empresario poderia tomar que o problema se tornava um pouco extenso para ser feito manualmente, então foi decido fazer um programa que formula-se o problema. Deste modo decidimos usar a linguagem JAVA para fazer o programa, que criaria as matrizes de transferencia e de custos com o recurso a loops e quando as matrizes estivessem feitas as usaria para resolver o problema.

1.2 Descrição da implementação do programa desenvolvido

O programa que nós desenvolvemos em JAVA consiste em 3 classes, a classe Mtransferencia que é responsavel por gerar todas as matrizes de probabilidades, a classe Mcusto que é responsavel por gerar todas as matrizes de custos e por fim a classe Main que após ter todas as matrizes de probabilidades e de custos geradas resolve o problema de programção dinamica estócastica.

1.2.1 Classe Mtransferencia

Esta classe como já foi dito está encarregue de gerar as matrizes de probabilidades. Isto foi possivel devido a indentificarmos os padrões consuante a diferença entre o estado final e inicial, ou seja diferença = estado final - estado inicial e a partir dai calcular todos os casos possiveis para o dado estado inicial–final. Vejamos um exemplo para explicar melhor visto que a classe é demasiado extença para mostrar tudo. Vamos imaginar o estado inicial 5 e o final 10.

A probabilidade para este par de estados, 5–10, sendo E(x) a probabilidade de ser entregue(s) x carro(s) numa dada filial e P(x) a probabilidade de ser pedido(s) x carros numa dada filial, é igual a
E(10)*P(5) // sendo os 5 pedidos satisfeitos.
+ E(10)*P(6) // 5 pedidos satisfeitos, 1 insatisfeito.
+ E(10)*P(7) // 5 pedidos satisfeitos, 2 insatisfeitos.
+ E(10)*P(8) // 5 pedidos satisfeitos, 3 insatisfeitos.
+ E(10)*P(9) // 5 pedidos satisfeitos, 4 insatisfeitos.
+ E(10)*P(10) // 5 pedidos satisfeitos, 5 insatisfeitos.
+ E(10)*P(11) // 5 pedidos satisfeitos, 6 insatisfeitos.
+ E(10)*P(12) // 5 pedidos satisfeitos, 7 insatisfeitos.
+E(5)*P(0) // casos particulares devido a diferença ser positiva
+E(6)*P(1)
+E(7)*P(2)
+E(8)*P(3)
+E(9)*P(4)
Resumindo, o algoritmo tem uma fase inicial em que faz os casos genericos em que a probabilidade de encomendas,E(), tem sempre o valor do estado final e a probabilidade de pedidos P(), vai desde o estado inicial até 12 que é o maior estado que existe.Depois tem uma segunda fase que depende da diferença entre estado e como neste caso é positiva,comçeamos o E() na diferença e vai incrementando até o valor do estado final, e o P() que começa em zero e vai incrementando. Este algoritmo que acabamos de explicar é usado para as matrizes em que a decisão é a de não transferir.

Para as restantes matrizes fazemos 1/2/3 shift(s) relativamente a matriz de não transferir que, como nós definimos inicialmente que os nossos estagios eram no inicio, são para a esquerda no caso das matrizes de receber carros e para a direita de transferir.Ou seja matriz de receber 3 carros = 3 shifts a esquerda e a matriz de transferir 3 carros = 3 shifts a direita em relação a matriz de não

transferir.

Depois de termos as 7 matrizes de probabilidades 13x13, uma para cada decisão, de cada matriz nós fazesmos a multiplicação das matrizes entre as filiais para obter as 7 matrizes de 169x169,ou seja
multiplicamos as matrizes de não transferir de ambas as filiais.
Multiplicamos a de receber 1 carro da filial 1 como a de transferir 1 carro da filial 2
Multiplicamos a de receber 2 carros da filial 1 como a de transferir 2 carros da filial 2 e assim sucessivamente até termos as 7 matrizes de 169 por 169 que estão declaradas como variaveis de instancia na classe para ser mais facil o acesso.

1.2.2 Classe Mcusto

. A classe Mcusto que gera todas as matrizes de custo opera num modo muito parecido com a da classe Mtransferencia a diferença é que no calculo, da matriz de decisão de não transferir, para cada posição a cada possibilidade encontrada multiplica-se 30x o numero de pedidos que são satisfeitos. Usando o exemplo anterior,5–10, o calculo fica do seguinte modo
30*5*E(1O)*P(5) + 30*5*E(1O)*P(6) + 30*5*E(1O)*P(7) +.....
Como sabemos que o estado inicial é 5 neste caso, então o maximo de pedidos satisfeitos é 5 e mesmo que sejam pedidos mais o numero que o 30 é multiplicado nunca sobe mais do que o estado inicial.
No final da matriz vamos retirar 10 todas as celulas cujo estado final seja maior do que 8 devido a taxa fixa do estacionamento.

No calculo das restantes matrizes de decisão tambem fizemos o shift das mesmas como fizesmos na Mtransferencia com a diferencia que nas matrizes de transferencia retiravamos 7 euros a cada shift, devido ao preço a pagar por cada transferencia de carros.

Por fim o calculo das 7 matrizes de 169 por 169 é igual ao calculo que usamos na classe Mtransferencia a emparelharmos
Não transferi filial 1 - Não transferir filial 2
receber 1 filial 1 - transferir 1 filial 2
receber 2 filial 1 - transferir 2 filial 2
receber 3 filial 1 - transferir 3 filial 2
transferir 1 filial 1 - receber 1 filial 2
transferir 2 filial 1 - receber 2 filial 2
transferir 3 filial 1 - receber 3 filial 2
e deste modo obtemos as 7 matrizes de custo necessarias para resolver o problema, estas matrizes tambem estão como variaveis de instancia na classe para ser mais facil o acesso.

1.2.3 Classe Main

Esta classe contem o metodo main() que inicia o programa e no metodo main o que acontece é que cria 2 objetos, um Mtransferencia e um Mcusto, e como esses objetos tem nas suas variaveis de instancia as 7 matrizes pode usa-los para todos os calculos.De seguida chama o metodo mkmodel em que passa esses mesmos objetos para a resolução do problema. Para resolver o problema usamos varios ciclos e no final na variavel Dn temos o nosso resultado.

1.3 Analise dos resultados

Os resultados que obtivemos ao longo de algumas iterações tende a estabilizar em 8, o que nos levou a desconfiar da veracidade do mesmo pois tinhamos ideia que devia dar um valor bastante superior e procuramos testar o nosso programa e chegamos a conclusão que se ouver um eventual erro será na geração das matrizes de custos visto que confirma-mos que as nossas matrizes de probabilidades estavam bem, a soma das linhas da sempre 1, e o algoritmo na main que resolve o problema está aparenta estár correto visto que o testamos com valores de matrizes de custo/probabilidade de problemas resolvidos nas aulas e chegava ao mesmo resultado que foi atingido na aula.Tentamos então procurar por um eventual erro na Mcusto, classe que gera as matrizes de custo, mas foi um esforço sem qualquer tipo de resultados.

```
7.977052871318392, 7.977052871317937, 7.977052871318392, 7.977052871319302,
7.97705287131771, 7.9770528713186195, 7.9770528713186195, 7.97705287131771,
7.977052871320211, 7.977052871319074, 7.977052871318165, 7.9770528713186195,
7.97705287131771, 7.977052871319074, 7.977052871319756, 7.977052871318847, 7
```

Figure 2: Algumas linhas do resultado do Dn

2 Parte 2 do trabalho pratico

Para esta fase do trabalho tendo em conta que consistia em escrever um resumo de um artigo cientifico achamos por melhor ficar num documento a parte, MEIO-Parte2.pdf, para melhor leitura do mesmo.

3 Conclusão

No nossa prespetiva este trabalho foi positivo visto que concluimos ambas as partes propostas no inunciado mas tambem ficamos com a certeza que podiamos ter feito melhor tanto na parte 1 como na 2.Achamos que o trabalho foi bastante trabalhoso principalmente na parte 1 visto que fazer debug do das matrizes de probabilidade/custo nem sempre era tarefa facil e dado ao tamanho das matrizes geradas era facil acontecer o erro no entanto acabamos este trabalho com a certeza que nos fez desenvolver varias competencias principalemente na resolução de problemas dinamicos estocasticos.

4 Anexos

4.1 Dados fornecidos pelo docente

Grupo que inclui o Aluno com o Nº 84220
MEIO-TP1 - Tabelas de probabilidades de pedidos e entregas de automóveis

FILIAL 1

Número de clientes:	0	1	2	3	4	5	6	7	8	9	10	11	12
Probabilidade (pedidos):	0.0564	0.1036	0.1496	0.1208	0.1184	0.1076	0.0976	0.0712	0.0620	0.0504	0.0312	0.0232	0.0080
Probabilidade (entregas):	0.0128	0.0636	0.1176	0.1780	0.2072	0.1564	0.1136	0.0708	0.0440	0.0208	0.0088	0.0044	0.0020

FILIAL 2

Número de clientes:	0	1	2	3	4	5	6	7	8	9	10	11	12
Probabilidade (pedidos):	0.0340	0.0724	0.1204	0.1456	0.1356	0.1240	0.1004	0.0828	0.0612	0.0512	0.0368	0.0284	0.0072
Probabilidade (entregas):	0.0224	0.0828	0.1788	0.2112	0.1836	0.1452	0.0920	0.0468	0.0264	0.0072	0.0024	0.0004	0.0008

Figure 3: Dados fornecidos

4.2 Matrizes de transição

Devido ao facto de as matrizes geradas serem demasiado grande para estarem em pdf decimos que eles vao estar em dois documentos excel Mcustos.xlsx e Mtransf.xlsx em que tem as matrizes geradas de custos e de probabilidades respetivamente.

4.3 Codigo do prgrama criado

Tambem devido a o programa criado ainda ter um codigo um pouco extenso devido aos ciclos para por em pdf decimos adicionar o codigo numa pasta ,src, em que pode encontrar todo o codigo criado pelo grupo para a geração do modelo.

4.4 Output do programa

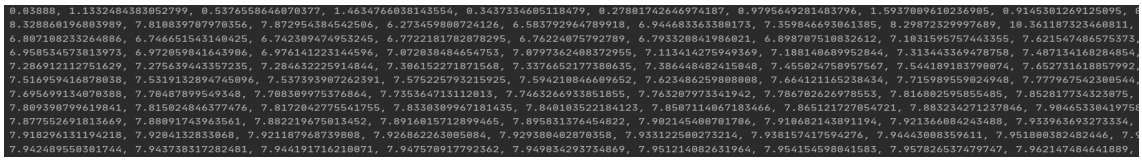

Figure 4: Output mais detalhado

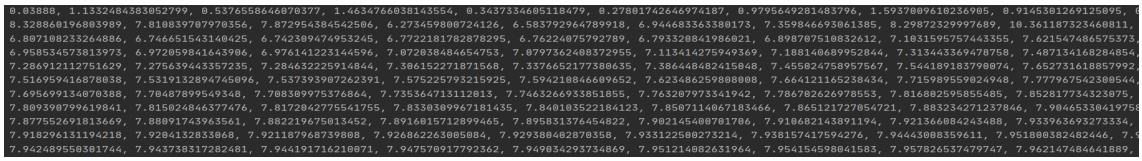

Figure 5: Output mais detalhado