

## Otimização II

### Trabalho Prático

**Professor:** André Rodrigues da Cruz

**E-mail:** andreacruz@timoteo.cefetmg.br

**Data de Entrega:** 26/11/2017

**Valor:** 40 pontos

### Instruções

- O trabalho prático deve ser feito em grupo de no máximo 4 integrantes.
- Não serão aceitos trabalhos entregues após a data estabelecida.
- Os algoritmos devem ser implementados em scripts “.m” que devem ser executados no GNU Octave. Cada função implementada deve estar em um arquivo separado, como também cada script que soluciona um problema. Para cada exercício, crie um diretório cujo nome será a numeração do mesmo. Deve-se estar presente nestes diretórios apenas os scripts que solucionam o problema.
- Os arquivos devem ser compactados no formato zip e submetidos no sistema AVA.
- Em cada script, deixe um cabeçalho comentado com os nomes e números de matrícula dos integrantes do grupo, e uma descrição da solução desenvolvida.
- Grupos que não seguirem as instruções acima, perderão automaticamente 5 pontos obtidos na nota obtida.
- Soluções cujo script não compila ou não funcionam corretamente receberão nota 0 no exercício.
- Grupos que copiarem, pelo menos parcialmente, alguma solução receberão nota 0 no trabalho.

### Problemas

- 1) Visando solucionar o problema de um jogo com 2 jogadores e soma 0, resolva:
  - a) (5 pontos) Implemente um script com uma função que receba uma matriz que representa uma tabela de prêmios e retorne, para cada jogador, o vetor de probabilidades de se selecionar cada estratégia, e o valor do jogo. Em outras palavras, uma função que resolva o problema via estratégias mistas. Use a função `glpk` para solucionar o problema linear.
  - b) (1 ponto) Usando a solução da questão anterior, implemente um script que solucione o jogo de par ou ímpar. Simule tal jogo 100 vezes e acumule os prêmios retornados.
  - c) (4 pontos) Usando a solução da questão anterior, implemente um script que:
    - (i) Resolva o jogo com a seguinte tabela de prêmios:

	$B_1$	$B_2$	$B_3$
$A_1$	3	-1	-3
$A_2$	-2	4	-1
$A_3$	-5	-6	-2

- (ii) Simule o jogo em 100 rodadas, com a solução ótima obtida, e acumule e apresente o somatório dos prêmios.
  - (iii) Simule o jogo em 100 rodadas, com uma solução um pouco distinta da ótima (a norma euclidiana da diferença das duas não passa de 0,01), e acumule e apresente o somatório dos prêmios.
  - (iv) Compare numericamente o somatório dos prêmios das duas simulações.
- 2) (10 pontos) Seja a cadeia de Markov representada pela matriz de transição apresentada abaixo. Suponha que a cadeia inicie no estado 1. Construa um script que simule 1000 vezes e estime, separadamente, os valores de probabilidades de a cadeia cair nos estados absorventes 0 e 4. Em cada simulação deve ser observado, no máximo, 100 transições.

$$\mathbf{P} = \begin{matrix} & \text{Estado} & 0 & 1 & 2 & 3 & 4 \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 2/3 & 0 & 1/3 & 0 & 0 \\ 0 & 2/3 & 0 & 1/3 & 0 \\ 0 & 0 & 2/3 & 0 & 1/3 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{matrix}$$

3) Visando simular sistemas de filas, resolva:

- a) (5 pontos) Implemente um script com uma função que receba a taxa de chegada,  $\lambda$  (número de pessoas por unidade de tempo), a taxa de atendimento,  $\mu$  (número de pessoas por unidade de tempo), o número de atendentes do sistema de filas,  $s$ , e o tempo total de simulação (número de iterações),  $T$ . Considere as taxas fornecidas como valores de probabilidades, em cada iteração da simulação, de modo que:
- Se em uma iteração, um valor aleatório entre 0 e 1 proveniente da distribuição uniforme for menor que  $\lambda$ , então um novo cliente é gerado e entra na fila de espera.
  - Quando um atendente fica ocioso, em uma iteração, ele receberá o primeiro da fila. Faça isso para cada atendente ocioso. Para cada atendente ocupado, em uma iteração, se um valor aleatório entre 0 e 1 proveniente da distribuição uniforme for menor que  $\mu$ , então o cliente corrente é considerado como atendido e sai do sistema. Se esta ação ocorrer, o atendente pode receber um novo cliente apenas na próxima iteração.

Use a estrutura de dados fila para organizar os clientes que aguardam atendimento. Retorne dois vetores, um que armazena o número de clientes que aguardam atendimento ao longo do tempo, e um que armazena o total de clientes no sistema ao longo do tempo.

- b) (5 pontos) Elabore um script que simule 100 vezes três cenários para o sistema de filas ao longo do tempo  $T = 1000$ , que são:
- $\lambda = 0,3$ ,  $\mu = 0,25$ ,  $s = 1$ .
  - $\lambda = 0,3$ ,  $\mu = 0,25$ ,  $s = 2$ .
  - $\lambda = 0,3$ ,  $\mu = 0,20$ ,  $s = 3$ .

Para cada cenário simulado, gere gráficos que descrevam o número de clientes total e aguardando atendimento ao longo do tempo. Não se esqueçam de inserir título e legendas adequadas.

- 4) (10 pontos) Visando analisar o comportamento sistemas de fila  $M/M/s$ , construa um script com uma função que receba a taxa de chegada,  $\lambda$  (número de clientes por minuto), e a taxa de atendimento,  $\mu$  (número de clientes por minuto). Para  $s \in \{1, 2, 3, 4\}$ , tal script deve apresentar uma tabela comparativa apresentando os valores das grandezas  $P_0, P_1, P_2, P_5, P_{10}, L, L_q, W, W_q, P(\mathcal{W}_q > 0), P(\mathcal{W}_q > 1), P(\mathcal{W}_q > 2)$  e  $P(\mathcal{W}_q > 5)$ .