

Estimando probabilidades por simulações computacionais

Lúcio S. Fassarella

Resumo

Discutimos a resolução de problemas probabilísticos mediante simulações computacionais de experimentos aleatórios. Para tanto, definimos o conceito de *algoritmo-solução probabilístico* e ilustramos a aplicação do método a seis problemas, apresentando seus algoritmos-soluções em um formato conveniente para serem programados em computador. A aplicação do método requer apenas conhecimentos elementares de programação, sendo acessível a estudantes do Ensino Médio. A proposta é motivada por três problemas clássicos da Probabilidade.

Palavras-chave: Algoritmo-solução; Resolução de Problemas; Experimentos Aleatórios; Probabilidade Geométrica; Método de Monte Carlo.

Abstract

We discuss the resolution of probabilistic problems through computer simulations of random experiments. For that, we define the concept of *probabilistic solution-algorithm* as well as illustrate the application of the method to six problems, presenting its solution-algorithms in a convenient format to be programmed in a computer. The application of the method requires only basic knowledge of programming, being accessible to high school students. The proposal is motivated by three classic Probability problems.

Keywords: Algorithm-solution; Resolution of Problems; Random Experiments; Geometric Probability; Monte Carlo Method.

1. Introdução

Nos primórdios da teoria da Probabilidade, Chevalier de Méré (1607-1684) procurou Blaise Pascal (1623-1662) para esclarecer uma questão que o intrigava:

Chevalier estava confuso com as probabilidades de dois jogos em voga nos cassinos da França naquela época. O primeiro consistia em lançar um dado quatro vezes e ganhar se tirasse pelo menos um 'ás' (número 1). O segundo consistia em lançar um par de dados 24 vezes e ganhar se tirasse um duplo ás (o número 1 em ambos os dados) pelo menos uma vez. O cavaleiro inculto achava que as probabilidades deviam ser iguais. (...) No entanto, na realidade, o primeiro jogo resultou em vitória com uma frequência um pouco maior do que no segundo jogo. Isso ficou conhecido como Paradoxo de Chevalier de Méré. (Aczel, 2007, p.39)

Também Jean le Rond d'Alembert (1717-1783) equivocou-se na resolução do seguinte problema elementar de probabilidade (Mlodinow, 2009, pp.59-60):

Problema 1. Em um lançamento simultâneo de duas moedas, qual é a probabilidade de o resultado ser cara e coroa?

Embora d'Alembert tenha sido um eminente matemático do século XVIII, podemos presumir que ele não tenha tentado verificar sua solução mediante a realização de experimentos: bastaria ter lançado um par de moedas algumas dezenas de vezes para perceber que a frequência relativa dos resultados com cara e coroa não é $1/3$ (que é o que ele havia concluído). Finalmente, Paul Erdős (1913-1996), um brilhante matemático do século XX, também se enganou em um problema elementar de probabilidade, o famoso problema de Monty Hall:

Problema 2. Em um programa de auditório, os participantes de um jogo recebem a opção de escolher uma dentre três portas, ganhando o que estiver atrás dela; atrás de uma há um carro e atrás das outras duas há cabras. O jogo ocorre em três etapas: primeiro, o participante escolhe uma das portas; depois o apresentador elimina uma das portas que esconde uma cabra dentre as outras duas; finalmente, o participante tem a opção de manter ou trocar definitivamente a porta escolhida pela porta que restou. Presumindo que o participante deseja ganhar o carro, a questão é saber o que lhe é mais vantajoso, manter ou trocar de porta?

Paul Erdős achava que tanto faz manter ou trocar a porta, pensando que a probabilidade de ganhar o carro é 50% em ambas alternativas. Ao ser informado que era mais vantajoso trocar de porta, ele,

um dos maiores matemáticos do século XX, afirmou: 'Impossível'. A seguir, quando apresentado a uma prova matemática formal da resposta correta, ainda assim não acreditou nela, e ficou irritado. Somente depois que um colega preparou uma simulação computadorizada na qual Erdős assistiu a centenas de testes que geraram um resultado de 2 para 1 a favor da mudança de escolha da porta, ele admitiu estar errado (Mlodinow, 2009, p.54)

Os casos de Chevalier de Méré, Jean d'Alembert e Paul Erdős sugerem que a aprendizagem de Probabilidade pode ser aprimorada pela realização de experimentos aleatórios, em que as probabilidades dos eventos podem ser estimadas pelas frequências relativas. Entretanto, realizar uma quantidade suficiente de experimentos para se obter boas estimativas das probabilidades pode ser uma tarefa demasiado tediosa ou mesmo impossível na prática. Uma alternativa natural para contornar essa dificuldade é usar computadores: em muitas situações, eles podem ser programados para realizar um número bastante grande de simulações de experimentos aleatórios.

Neste artigo, mostramos como utilizar simulações computacionais de experimentos aleatórios para obtenção de estimativas para probabilidades.

Na Seção 2 definimos o conceito de *algoritmo-solução probabilístico* para fundamentar a ideia. Na Seção 3 apresentamos alguns problemas envolvendo o cálculo de probabilidades e apresentamos resoluções algorítmicas à guisa de exemplos ilustrativos. Na Seção 4 tecemos algumas considerações finais.

2. Algoritmo-solução probabilístico

Vamos definir alguns conceitos úteis para a discussão, seguindo ideias apresentadas em [5].

Definição 1. Algoritmo-solução de um problema matemático é um algoritmo cuja execução gera a solução exata ou uma aproximação arbitrariamente precisa da solução exata.

Em outras palavras, um algoritmo-solução de um problema gera sua solução com erro inferior a um limite preestabelecido arbitrariamente. Para além da questão de saber se um determinado problema matemático possui algum algoritmo-solução, temos que reconhecer que essa definição é demasiada estrita para admitir que simulações computacionais de experimentos aleatórios possam genericamente constituir algoritmos-soluções de problemas probabilísticos. A razão disso é que, em geral, existem realizações de experimentos aleatórios que geram frequências relativamente significativamente diferentes das probabilidades corretas. (Por exemplo, podemos tentar obter aproximações para as probabilidades associadas aos possíveis resultados do lançamento de uma moeda realizando uma sequência de 100 lançamentos e calculando as frequências relativas de cada possibilidade, cara ou coroa. Mesmo que a moeda não seja viciada, é possível obter sequências de resultados em que as frequências relativas são significativamente diferentes de 50%.) Essa circunstância torna necessário flexibilizar a definição de algoritmo-solução para admitir probabilidades:

Definição 2. Algoritmo-solução probabilístico de um problema matemático é um algoritmo cuja execução gera uma aproximação arbitrariamente precisa da solução exata do problema com probabilidade arbitrariamente próxima de 1, quaisquer que sejam os valores previamente dados para essas aproximações.

A Lei dos Grandes Números (LGN) garante que a Definição 2 é suficiente para que os problemas de probabilidade típicos possuam algoritmos-soluções probabilísticos. Para justificar e esclarecer isso, apresentamos, em uma forma conveniente, a versão fraca da LGN e recomendamos [7] para detalhes acerca da terminologia, notação e outras versões da LGN.

Teorema 1 (Lei Fraca dos Grandes Números de Bernoulli). *Considere um espaço amostral Ω munido de uma probabilidade \mathcal{P} , e seja $A \subseteq \Omega$ um evento com probabilidade $\mathcal{P}(A) = p \in [0, 1]$ de sucesso em um ensaio de Bernoulli. Para uma sequência infinita de ensaios independentes desse tipo, seja $\{X_n; n = 1, 2, \dots\}$ a sucessão dos resultados da ocorrência de A , ou seja,*

$$X_n = \begin{cases} 1, & \text{se } A \text{ ocorrer no } n\text{-ésimo ensaio;} \\ 0, & \text{se } A \text{ não ocorrer no } n\text{-ésimo ensaio.} \end{cases}$$

Então, a proporção das ocorrências do evento A converge em probabilidade para p , i.e.,

$$\lim_{n \rightarrow \infty} \mathcal{P} \left(\left| \frac{X_1 + \dots + X_n}{n} - p \right| > \varepsilon \right) = 0, \quad 8\varepsilon > 0.$$

Dizendo em palavras, isso significa que a probabilidade de o valor absoluto da diferença entre ‘a probabilidade teórica p do evento A ocorrer’ e a ‘proporção de ocorrências do evento A em uma sequência finita de ensaios independentes’ ser maior do que uma predeterminada margem de erro $\varepsilon > 0$ pode ser feita arbitrariamente próxima de zero realizando-se um número suficientemente grande desses ensaios.

Observação 1. Embora possamos dizer elusivamente que a LGN estabelece que probabilidade teórica de um evento aleatório pode ser estimada com precisão arbitrária pela proporção das ocorrências do evento em um número suficientemente grande de ensaios independentes, fato é que ela não garante a precisão dessa estimativa com certeza – o que explica o enunciado da Definição 2. Assim, por exemplo, a LGN não garante que a probabilidade de obtermos cara no lançamento de uma moeda

equilibrada será certamente mais bem estimada pelos resultados de 100 mil lançamentos do que pelos resultados de apenas 100 lançamentos; em vez disso, ela diz que a probabilidade de o valor 0.5 e a estimativa obtida pela observação de lançamentos diferirem em mais de, *e.g.*, 0.1 será menor no caso de 100 mil lançamentos do que no caso de 100 lançamentos.

Teoricamente, precisamos recorrer à Estatística para saber quantos ensaios aleatórios devem ser realizados para estimar (com determinado grau de certeza) a probabilidade de um evento com precisão previamente estabelecida. Todavia, podemos deixar a Estatística de lado em situações simples, bastando verificar a consistência dos resultados comparando várias estimativas. Interessados em obter detalhes sobre a análise de erros recomendamos [4], onde é abordado o cálculo de probabilidade geométrica de modo semelhante ao ilustrado no Problema 5 da próxima seção.

Para especialistas, deve estar claro que o método proposto constitui uma versão do Método de Monte Carlo:

[O Método de Monte Carlo é] qualquer método que resolve um problema gerando de uma quantidade adequada de números randômicos e observando as frações dos números que obedecem determinadas propriedades. (WEISSTEIN, 2016, tradução nossa)

O Método de Monte Carlo é ocasionalmente ensinado nas disciplinas introdutórias ao Cálculo Diferencial e Integral dos cursos de graduação em uma acepção específica para o cálculo de áreas de regiões planas, de uma forma que pode ser enquadrada na chamada *probabilidade geométrica*. Entretanto, a ideia é suficientemente simples para que possa ser abordada já na Educação Básica.

Na próxima seção apresentamos algoritmos-soluções probabilísticos para alguns problemas de probabilidade na perspectiva de ilustrar a ideia, incluindo alguns de probabilidade geométrica.

3. Problemas e Algoritmos-soluções probabilísticos

Nesta seção apresentamos alguns problemas probabilísticos e algoritmos-soluções, começando pelos problemas 1 e 2 já mencionados. Os algoritmos são relativamente fáceis de entender, mas explicamos a dinâmica de cada um para que leitores menos familiarizados possam compreender o processo de elaboração e codificação. Acreditamos que os exemplos são suficientes para que os leitores possam compreender o processo de simulação de experimentos aleatórios e se habilitem a resolver muitos outros problemas.

Os algoritmos-soluções são escritos em pseudocódigo, com a indentação denotando a estrutura ou hierarquia de comandos – como na linguagem Python [8]. Todos eles foram programados em Python 3.4.3, e os resultados das suas execuções são apresentados junto com as respostas teóricas de cada problema.

Para simplificar a discussão e a escrita dos algoritmos-soluções, consideramos a seguinte notação:

Definição 3. Se S for um conjunto finito, $\text{Random}(S)$ denota a função para escolha aleatória de um elemento de S , considerando que todos os elementos de S são igualmente prováveis. Se S for um intervalo limitado de \mathbb{R} , $\text{Random}(S)$ denota a função que gera um número aleatório em S segundo a distribuição de probabilidade uniforme.

Começamos apresentando algoritmos-soluções para os problemas 1 e 2.

A ideia do algoritmo-solução probabilístico para o Problema 1 consiste em realizar uma sequência de simulações do lançamento de duas moedas e contar o número de vezes em que os resultados dão cara e coroa (diferentes). A razão entre esse número e o total de lançamentos simulados dar-nos-á uma aproximação da resposta do problema (a probabilidade do lançamento de duas moedas resultar em cara e coroa).

Assim, no pseudocódigo a seguir armazenamos os resultados de cada lançamento do par de moedas nas variáveis a e b , representando as faces cara e coroa pelos números 0 e 1, respectivamente. Usamos a variável c para contar o número de experimentos simulados que resultam em cara e coroa, sendo inicializada em 0 (linha 2). Cada simulação de lançamento consiste em atribuir às variáveis a e b respectivos resultados de $\text{Random}(\{0, 1\})$ (linhas 4 e 5); o algoritmo compara esses resultados e soma uma unidade à variável c se forem diferentes (linhas 6 e 7). A variável N especifica o número de simulações de lançamento das moedas, fixado em 1000 (linha 1) mas poderia ser qualquer outro natural.

Algoritmo-solução probabilístico do Problema 1

```
1:  $N = 1000$   
2:  $c = 0$   
3: Repita  $N$  vezes:  
4:    $a = \text{Random}(\{0, 1\})$   
5:    $b = \text{Random}(\{0, 1\})$   
6:   Se  $a \neq b$  então:  
7:      $c = c + 1$   
8: Retorne  $c/N$ 
```

A execução desse algoritmo-solução retornou o número 0.500382. O resultado teórico é exatamente 0.5.

No Problema 2 temos que verificar qual estratégia é mais vantajosa dentre as duas em questão:

ntp: a estratégia de *nunca trocar de porta*;

stp: a estratégia de *sempre trocar de porta*.

Analisando o Problema 2, podemos deduzir que o participante que adota a estratégia ntp ganha o carro em uma rodada se, e somente se, ele acerta na primeira escolha a porta atrás da qual está o carro; enquanto o participante que adota a estratégia stp ganha o carro em uma rodada se, e somente se, ele erra na primeira escolha a porta atrás da qual está o carro. Percebemos que, para ambas as estratégias, o resultado de ganhar ou perder o carro em uma rodada independe da porta eliminada pelo apresentador. Entendemos que a vantagem de uma estratégia em relação à outra é dada pela proporção de ganhos: a estratégia que tiver a maior proporção de ganhos é a vantajosa.

Partindo dessas constatações, o algoritmo-solução probabilístico para o Problema 2 consiste em simular uma sequência de rodadas e contar o número de vezes em que cada estratégia ganha o carro. Para cada estratégia, a razão entre o número de ganhos e o total de simulações dará sua probabilidade de sucesso. No pseudocódigo a seguir, a variável N determina o número de simulações (fixada em 1000 na linha 1); as portas são representadas pelos números 1, 2 e 3; as variáveis ntp e stp contam o número de simulações em que a respectiva estratégia resulta em ganhar o carro, sendo inicializadas em 0 (linhas 2 e 3). Em cada simulação, “carro” denota a escolha aleatória da

porta atrás da qual está o carro (linhas 5) e “porta” denota a porta escolhida inicialmente pelo participante (linhas 6); o algoritmo verifica se a porta escolhida é a mesma que tem o carro e, caso positivo, acrescenta uma unidade à estratégia ntp (linhas 7 e 8); bem como verifica se a porta escolhida é diferente da que tem o carro e, caso verdadeiro, acrescenta uma unidade à estratégia stp (linhas 9 e 10). O algoritmo retorna às proporções de sucesso de cada estratégia, a partir das quais descobrimos qual é vantajosa (linha 11).

Algoritmo-solução probabilístico do Problema 2

```

1: N = 1000
2: ntp = 0
3: stp = 0
4: Repita N vezes:
5:   carro = Random ({1, 2, 3})
6:   porta = Random ({1, 2, 3})
7:   Se porta = carro então:
8:     ntp = ntp + 1
9:   Se porta ≠ carro então:
10:    stp = stp + 1
11: Retorne ntp/N e stp/N

```

A execução desse algoritmo-solução retornou para a estratégia ntp o número 0.333497 e para a estratégia stp o número 0.666503. Teoricamente, a proporção de ganhos da estratégia ntp é exatamente $1/3$, enquanto a proporção de ganhos da estratégia stp é exatamente $2/3$. Concluimos que a estratégia stp é mais vantajosa do que a estratégia ntp, na média resultando no dobro de ganhos.

O Problema 3 a seguir foi apresentado e resolvido analiticamente em [11]:

Problema 3. Duas pessoas decidem se encontrar em um determinado local entre 11 e 12 horas, com a combinação prévia de que a primeira pessoa a chegar esperará no máximo 15 minutos pela outra. Ache a probabilidade de o encontro acontecer nessas circunstâncias, admitindo que ambas as pessoas chegam ao local em instantes aleatórios compreendidos entre 11 e 12 horas.

Esse é um problema de probabilidade geométrica [12], cuja resolução requer apenas o princípio de que a probabilidade de um ponto escolhido ao acaso em um intervalo finito $U \subset \mathbb{R}$ pertencer a um subintervalo $E \subseteq U$ é dada pela razão entre a medida (comprimento) de E pela medida (comprimento) de U . A condição necessária e suficiente para que o encontro aconteça é que os instantes de chegada de cada uma das pessoas no intervalo de 60 minutos entre 11 e 12 horas difiram por, no máximo, 15 minutos (não importando quem chega primeiro).

Assim, no seguinte algoritmo-solução probabilístico N designa o número de experimentos aleatórios simulados, a e b representam os instantes de chegada das pessoas a partir das 11 horas (com o tempo medido em minutos), e c conta o número de experimentos nos quais ocorrem encontros (*i.e.*, $|a-b| \leq 15$). Em cada simulação, os instantes de chegada das pessoas a e b são dados pela função $\text{Random}([0, 60])$ (linhas 4 e 5); o algoritmo verifica se a diferença é menor ou igual a 15 (linha 6) e acrescenta uma unidade à variável c caso isso seja verdadeiro (linha 7). O algoritmo retorna o quociente entre c , o número de encontros obtidos, e N , o total de simulações.

Algoritmo-solução probabilístico do Problema 3

- 1: $N = 1000$
- 2: $c = 0$
- 3: Repita N vezes:
 - 4: $a = \text{Random}([0, 60])$
 - 5: $b = \text{Random}([0, 60])$
 - 6: Se $|a - b| \leq 15$ então:
 - 7: $c = c + 1$
- 8: Retorne c/N

A execução desse algoritmo-solução retornou o número 0.437887. Teoricamente, a probabilidade de o encontro acontecer é exatamente $7/16 = 0.4375$.

O Problema 4 a seguir é uma adaptação do primeiro problema apresentado em [6], onde o leitor encontra uma definição do conceito de probabilidade condicional e diversos problemas contextualizados em um jogo similar ao apresentado aqui:

Problema 4. Considere o seguinte jogo que consiste no lançamento de um par de dados e tem a seguinte regra de pontuação:

- se caírem duas faces diferentes de 4, então o jogador não marca pontos;
- se cair uma face 4, então a pontuação do jogador é igual ao valor que saiu na outra face.

Qual é a probabilidade de um jogador marcar 3 pontos em uma jogada, sabendo-se que um dos dados caiu na face 4?

A ideia básica para elaborar um algoritmo-solução probabilístico para um problema de probabilidade condicional com base em simulações de experimentos aleatórios consiste em:

- i) simular experimentos aleatórios no espaço amostral dos resultados possíveis;
- ii) identificar e contar os eventos válidos, ou seja, aqueles que cumprem as condicionantes do problema – no caso, a condicionante é que pelo menos um dos dados caia com a face 4;
- iii) identificar e contar os eventos favoráveis, ou seja, os eventos válidos para os quais buscamos determinar a probabilidade de ocorrerem – no caso, um dado cai com a face 4 e o outro com a face 3;
- iv) estimar a probabilidade condicional dos eventos favoráveis, calculando o quociente entre o número de eventos favoráveis pelo número de experimentos válidos.

No seguinte algoritmo-solução probabilístico para o Problema 4, são realizadas simulações para o lançamento de um par de dados representados pelas variáveis a e b . A variável cv conta as ocorrências dos eventos válidos (caracterizados na linha 7) e a variável cf conta as ocorrências dos eventos favoráveis (caracterizados nas linhas 9 e 11). O algoritmo realiza $N = 1000$ simulações e retorna o quociente entre o número de eventos favoráveis pelo número de eventos válidos (linha 13).

Algoritmo-solução probabilístico do Problema 4

```

01: N = 1000
02: cv = 0
03: cf = 0
04: Repita N vezes:
05:     a = Random ({1, 2, 3, 4, 5, 6})
06:     b = Random ({1, 2, 3, 4, 5, 6})
07:     Se a = 4 ou b = 4, então:
08:         cv = cv + 1
09:     Se a = 4 e b = 3, então:
10:         cf = cf + 1
11:     Se a = 3 e b = 4, então:
12:         cf = cf + 1
13: Retorne cf/cv
    
```

A execução desse algoritmo retornou o número 0.16279069767441862. A resposta teórica é $2/11 = 0.181818\dots$

O Problema 5 ilustra o cálculo de áreas de regiões planas usando simulações computacionais de eventos aleatórios. Foi apresentado e resolvido de modo similar em [11] e [4].

Problema 5. Calcule a área delimitada pelo círculo de raio unitário.

Como a área do círculo de raio unitário é π , a resolução computacional do problema dá-nos uma estimativa do seu valor.

Para resolver o problema mediante a realização de experimentos aleatórios, basta considerar no plano cartesiano um círculo de raio unitário e um retângulo que o contenha: assim, a probabilidade de que um ponto escolhido ao acaso no retângulo pertença ao círculo é igual à razão entre a área do círculo e área do retângulo. A Figura 1 ilustra a ideia no caso em que o retângulo escolhido é o quadrado de lado unitário (usado no algoritmo-solução a seguir).

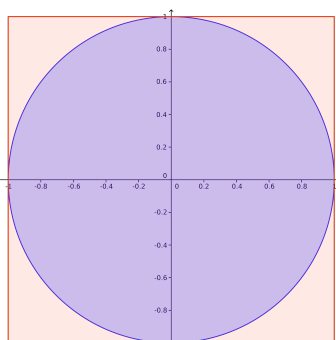


Figura 1: **Círculo contido em um quadrado.**

Fonte: Próprio autor.

Se o leitor acompanhou os exemplos apresentados anteriormente, provavelmente achará natural e

fácil de compreender o seguinte algoritmo-solução:

Algoritmo-solução probabilístico do Problema 5

- 1: $N = 1000$
- 2: $c = 0$
- 3: Repita N vezes:
 - 4: $x = \text{Random}([-1, 1])$
 - 5: $y = \text{Random}([-1, 1])$
 - 6: Se $x^2 + y^2 \leq 1$, então:
 - 7: $c = c + 1$
- 8: Retorne $4c/N$

A execução do algoritmo retornou o número 3.140976, enquanto o valor de π até a sexta casa decimal é 3.141592.

O próximo problema foi proposto e resolvido em [2], num contexto envolvendo *Cadeias de Markov*. Com isso vemos que a simulação de experimentos aleatórios aplica-se a um amplo espectro de situações, embora possa requerer maior trabalho de programação.

Problema 6. Uma seguradora possui três tipos de seguros (A, B, C) para automóveis no primeiro momento de vendas. As probabilidades de escolha desses pacotes são: 50% para a opção A, 30% para a opção B e 20% para a opção C, sendo obrigatório que o cliente use o mesmo tipo de seguro por pelo ou menos um mês. Após o primeiro mês de contrato foram constatadas as seguintes informações:

- Das pessoas que escolheram a opção A, 10% migram para a opção B e 20% para a opção C;
- Das pessoas que escolheram a opção B, 40% migram para a opção A e 10% para a opção C;
- Das pessoas que escolheram a opção C, 10% migram para a opção A e 10% para a opção B.

Se essa mesma seguradora quer saber qual a probabilidade de um cliente, após o primeiro mês, migrar para as opções de seguro A, B e C, o que deve ser feito?

Para simular computacionalmente o comportamento dos clientes da seguradora, com suas diferentes probabilidades para escolher ou trocar de seguro, vamos usar a função *Random* no intervalo $[0, 1]$ e associar as probabilidades de cada opção a subintervalos de $[0, 1]$ com comprimentos proporcionais a essas probabilidades. Para prevenir eventuais dúvidas, destacamos que a probabilidade de um intervalo independe de ele ser aberto, fechado ou semiaberto, fato que é coerente com a ideia de que um ponto isolado possui probabilidade zero. (Observamos que as probabilidades desse problema também poderiam ser simuladas em um conjunto discreto. Todavia, usando um intervalo enquadramos o problema nos termos de uma probabilidade geométrica. Além dessa abordagem nos ter proporcionado um algoritmo-solução simples, ela também possui a flexibilidade de nos permitir modificar as probabilidades sem mudar o espaço amostral.) Explicitamente, para simular a escolha no primeiro momento de vendas (50% para a opção A, 30% para a opção B e 20% para a opção C), associamos o intervalo $[0, 0.5]$ à opção A, $[0.5, 0.8]$ à opção B e $[0.8, 1]$ à opção C. Veja a ilustração na Figura 2. Uma vez simulada a primeira escolha, procedemos similarmente para simular a segunda escolha considerando as probabilidades de troca em cada caso.

No algoritmo-solução probabilístico a seguir, N designa o número de simulações; contaA , contaB e contaC contam os números de vezes em que a segunda escolha cai, respectivamente, na opção A, opção B ou opção C; o parâmetro s simula a primeira escolha e o parâmetro t simula a segunda escolha. Para tornar o algoritmo mais claro, inserimos comentários explicativos na cor verde

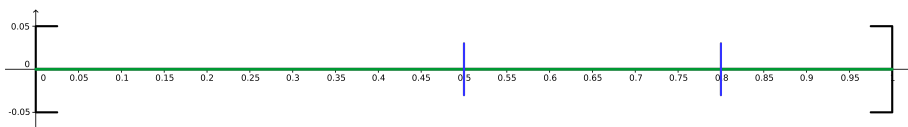


Figura 2: **Partição do intervalo $[0, 1]$ em três subintervalos.**

Fonte: Próprio autor.

segundo o símbolo “#”. O algoritmo retorna as probabilidades de um cliente optar por cada seguro após o primeiro mês, permanecendo com ele ou migrando de outro.

O leitor poderá perceber que o algoritmo pode ser simplificado, pois nele uma estrutura é repetida três vezes. Essa simplificação pode ser feita de várias maneiras, mas optamos por não fazer presumindo que isso dificultaria o entendimento.

Algoritmo-solução probabilístico do Problema 6

```

01: N = 10000
02: contaA = 0
03: contaB = 0
04: contaC = 0
05: Repita N vezes:
06:   s = Random ([0, 1])
07:   t = Random ([0, 1])
08:   Se  $0 \leq s < 0.5$  então: # primeira escolha na opção A
09:     Se  $0 \leq t < 0.1$  então: # troca para a opção B
10:       contaB = contaB + 1
11:     Se  $0.1 \leq t < 0.1 + 0.2$  então: # troca para a opção C
12:       contaC = contaC + 1
13:     Se  $0.1 + 0.2 \leq t \leq 1.0$  então: # permanência na opção A
14:       contaA = contaA + 1
15:   Se  $0.5 \leq s < 0.5 + 0.3$  então: # primeira escolha na opção B
16:     Se  $0 \leq t < 0.4$  então: # troca para a opção A
17:       contaA = contaA + 1
18:     Se  $0.4 \leq t < 0.4 + 0.1$  então: # troca para a opção C
19:       contaC = contaC + 1
20:     Se  $0.4 + 0.1 \leq t \leq 1.0$  então: # permanência na opção B
21:       contaB = contaB + 1
22:   Se  $0.5 + 0.3 \leq s \leq 1.0$  então: # primeira escolha na opção C
23:     Se  $0 \leq t < 0.1$  então # troca para a opção A:
24:       contaA = contaA + 1
25:     Se  $0.1 \leq t < 0.1 + 0.1$  então: # troca para a opção B
26:       contaB = contaB + 1
27:     Se  $0.1 + 0.1 \leq t \leq 1.0$  então: # permanência na opção C
28:       contaC = contaC + 1
29: Retorne contaA/N, contaB/N, contaC/N
    
```

A execução do algoritmo retornou as seguintes probabilidades para cada seguro ser escolhido por um cliente após o primeiro mês: seguro A: 0.4725; seguro B: 0.2179; seguro C: 0.3096. Os valores

teóricos são: seguro A: 0.49; seguro B: 0.22; seguro C: 0.29 [2].

4. Conclusão

Simulações computacionais de experimentos aleatórios podem ser usadas proficuamente no ensino e aprendizagem de Probabilidade, especialmente quando os próprios estudantes elaboram, programam e executam as simulações [5]. Aqui apresentamos uma estratégia geral para resolução de problemas probabilísticos que pode colaborar com outras abordagens, ampliando e aprofundando a experiência dos estudantes com os conceitos envolvidos nos problemas. Naturalmente, a ideia pode ser mais fácil ou mais difícil de ser aplicada dependendo da situação.

Evidentemente, elaborar e programar algoritmos-soluções para problemas matemáticos requer familiaridade com o assunto; mas é razoável supor que isso em breve tornar-se-á pelo menos tão comum nas aulas de Matemática quanto hodiernamente é comum usar planilhas eletrônicas e *softwares* de geometria dinâmica. Apontam nessa direção relatos de experiências bem-sucedidas, tais como [10].

O uso de experimentos aleatórios na resolução de problemas de probabilidade já foi abordado em alguns artigos publicados. Por exemplo, o Professor Nelson Tunala reportou em [11] os resultados de seis experiências físicas relacionados ao Problema da Agulha de Buffon, através das quais foram obtidas aproximações para o valor de π ; o Professor Paulo C. P. Carvalho mostrou em [3] como usar uma planilha eletrônica para estimar o valor esperado de uma variável aleatória mediante simulações computacionais; em [5], discutimos um problema de cálculo de valor esperado mediante simulações computacionais numa situação de probabilidade geométrica.

Por fim, deixamos ao leitor o desafio de resolver o Paradoxo de Chevalier de Méré apresentado na Introdução, usando uma abordagem algébrica e também a simulação computacional de experimentos aleatórios.

Referências

- [1] Aczel, A. D. *Quais são suas chances?* Rio de Janeiro: BestSeller, 2007.
- [2] Amorim, A. M. N.; Trindade, A. K. B.; Araujo Junior, F. P. S. *Cadeias de Markov: Uma introdução apresentável ao ensino médio com aplicação ao soneto “Amor é fogo que arde sem se ver”*. Professor de Matemática Online, v.8, n°2, 2020. DOI:10.21711/2319023x2020/pmo818.
- [3] Carvalho, P. C. P. *O Valor Esperado de uma Quantidade Aleatória*. RPM, v.89, pp.18-23, 2016.
- [4] Cavalcante, Igor Dutra. *Probabilidade Geométrica: Uma abordagem através do Método de Monte Carlo e Geometria Fractal*. Dissertação de Mestrado. Universidade Federal de Alagoas. Maceió, 2017.
- [5] Fassarella, L. S. *Resolução Computacional de Problemas de Probabilidade*. In: CMAC-SE, III, 2015, Vitória-ES. Proceedings Series of the Brazilian Society of Computational and Applied Mathematics, v.3, n°2, p.020129 [6 páginas], XXX. Disponível em: <<https://proceedings.sbm.ac.org.br/sbm.ac/article/view/1020/1033>>. Acesso em: 30 de julho de 2020.
- [6] Lopes, J. M. *Probabilidade condicional por meio da resolução de problemas*. RPM, v.62, pp.34-38, 2007. Disponível em: <<http://rpm.org.br/cdrpm/62/8.html>>. Acesso em: 30 de julho de 2020.
- [7] Magalhães, M.N. *Probabilidade e variáveis aleatórias*. 2ª. edição. São Paulo: Edusp, 2006.

- [8] Menezes, N. N. C. *Introdução à Programação com Python: Algoritmos e Lógica de Programação*. 3ª edição. São Paulo: Novatec Editora, 2019.
- [9] Mlodinow, L. *O Andar do Bêbado: como o acaso determina nossas vidas*. Rio de Janeiro: Zahar, 2009.
- [10] Souza, L. O.; Lopes, C.E. *O Uso de Simuladores e a Tecnologia no Ensino da Estocástica*. Bolema, v.24, n°40, pp.659-677, 2011. Disponível em: <<http://www.periodicos.rc.biblioteca.unesp.br/index.php/bolema/article/view/5287>>. Acesso em: 30 de julho de 2020.
- [11] Tunala, N. *Determinação de Probabilidades por Método Geométrico*. RPM, v.20, pp.16-22, 1992. Disponível em: <<http://rpm.org.br/cdrpm/20/5.htm>>. Acesso em: 30 de julho de 2020.
- [12] Wagner, E. *Probabilidade Geométrica*. RPM, v.34, pp.20-24, 1997. Disponível em: <<http://rpm.org.br/cdrpm/34/6.htm>>. Acesso em: 30 de julho de 2020.
- [13] Weisstein, E. W. *Monte Carlo Method*. Disponível em: <<http://mathworld.wolfram.com/MonteCarloMethod.html>>. Acesso em: 30 de julho de 2020.

Lúcio S. Fassarella
Universidade Federal do Espírito Santo
<lucio.fassarella@ufes.br>

Recebido: 30/07/2020
Publicado: 26/02/2021