**Revista Eletrônica
Paulista de Matemática**

ISSN 2316-9664
Volume 11, dez. 2017

Rogério César dos Santos

Membro dos grupos PI / UnB e
GIEM – Grupo de Investigação
em Educação Matemática UnB.
FUP/UnB
professorrogeriocesar@gmail.com

Antônio Luiz de Melo

FUP/UnB
almelo@gmail.com

O problema do ponto mais visitado em retângulos e paralelepípedos: casos particulares e conjecturas

The problem of the most visited point in rectangles and
parallelepipeds: particular cases and conjectures

Resumo

O Problema do Ponto Mais Visitado em quadrados foi resolvido por Santos e Castilho (2013), já em retângulos ou paralelepípedos o problema encontra-se em aberto. Este artigo pretende demonstrar uma solução do referido problema para casos particulares de retângulos e estabelecer conjecturas para os retângulos e os paralelepípedos em geral. A metodologia usada na prova dos casos particulares dos retângulos baseou-se no princípio de indução e nas ferramentas da análise combinatória. Para o estabelecimento das conjecturas nos casos gerais foram utilizadas simulações numéricas nos softwares livres Maxima e Python. A conclusão que chegamos ao fim do artigo é que é possível conjecturar que o ponto do plano mais visitado por caminhos em todos os retângulos do tipo M por N , com $M > N$, é o ponto $(1,0)$, já para os paralelepípedos, depende de suas dimensões. Também concluímos que é possível, por meio de simulações, conectar estudantes de matemática a problemas atualmente em aberto, levando-os ao entendimento do método investigativo e instigando-os à pesquisa matemática.

Palavras-chave: Ponto mais visitado. Soluções numéricas. Análise Combinatória. Caminhos no plano. Caminhos no espaço.

Abstract

The Problem of the Most Visited Point in squares was solved by Santos and Castilho (2013). Nevertheless, in rectangles or parallelepipeds, the problem is open. This article aims to demonstrate a solution for this problem for particular cases of rectangles and to establish conjectures for rectangles and parallelepipeds in general. The methodology used in the proof of the particular cases of the rectangles was based on the principle of induction and on the tools of combinatorial analysis. For the establishment of conjectures in general cases, numerical simulations were used in Maxima and Python free software programs. The conclusion that we reached at the end of the article is that it is possible to conjecture that the point of the plane most visited by paths in all $M \times N$ rectangles, with $M > N$, is the point $(1,0)$. For parallelepipeds, it depends on their dimensions. We also concluded that it is possible, through simulations, to connect mathematics students to currently open problems, by leading them to understand the investigative method and by instigating them to mathematical research.

Keywords: Most visited point. Numerical solutions. Combinatorial Analysis. Paths in the plane. Paths in space.

1 Introdução

Considerando caminhos que saem da origem $P = (0, 0)$ com destino em $Q = (M, N)$, com $M > N$, caminhos estes indo *sempre para cima ou para a direita*, deseja-se determinar por qual ponto (x, y) de coordenadas inteiras passam mais percursos (exceto, naturalmente, os pontos P e Q , que são de passagem obrigatória).

Figura 1 : Um caminho possível da origem até $Q = (5, 2)$. Por qual ponto passam mais caminhos?

Um contexto possível para este problema seria: se um vendedor de amendoins deseja fixar o seu local de venda em um ponto onde passa a maior quantidade de carros possível, numa tal rede rodoviária retangular, qual seria o ponto ideal a ser escolhido pelo vendedor, para que possa vender em maior quantidade? É o que será desenvolvido neste artigo.

1.1 O problema do ponto mais visitado em quadrados

Santos e Castilho (2013) mostramos que, para pontos de chegada do tipo $Q = (M, M)$ com $M > 1$, isto é, em circuitos *quadrados*, o ponto mais visitado é o ponto $(1, 1)$, juntamente com seu simétrico $(M - 1, M - 1)$, independentemente do valor de $M > 1$:

Figura 2: O ponto mais visitado em quadrados são os pontos $(1,1)$ e $(M - 1, N - 1)$, em destaque na figura

A solução do problema do ponto mais visitado em quadrados fora também explorada por Melo e Santos (2014), Santos (2016a) e Santos (2016b). No primeiro destes trabalhos, os autores descobriram novas propriedades do Triângulo de Pascal a partir da solução do problema dos percursos em quadrados. No segundo trabalho, o problema dos caminhos em quadrados é resolvido por meio das cadeias de Markov. No terceiro trabalho o problema dos percursos em quadrados é visto sob o ponto de vista da probabilidade.

Neste artigo, generalizamos este problema para retângulos e paralelepípedos. Faremos conjecturas baseadas em simulações nos softwares Python e Maxima. Trata-se de um problema que pode ser aplicado para alunos licenciandos ou bacharelados em matemática, despertando, quem sabe, o desejo por obter uma prova para este problema ainda em aberto para estes casos.

2 Retângulos até a ordem 150 por 149: abordagem numérica no software Maxima

Primeiramente, observemos que, como visto no trabalho de Santos e Castilho (2013), o número $c(x, y, M, N)$ de caminhos que passam por um ponto (x, y) em um retângulo M por N é:

$$c(x, y, M, N) = \frac{(x+y)!}{x!y!} \cdot \frac{(M-x+N-y)!}{(M-x)!(N-y)!} \quad (1)$$

O primeiro fator do produto acima corresponde ao número de caminhos que saem da origem $(0,0)$ e chegam ao ponto (x, y) . O segundo fator corresponde ao número de caminhos que saem de (x, y) e chegam ao ponto final (M, N) .

Usando esta fórmula, utilizei o software algébrico livre Maxima (ANDRADE, 2012) para a obtenção dos resultados de $c(x, y, M, N)$ para todos os valores possíveis de M , N , x e y , com M de 2 até 150 e N de 1 até 149, sempre com $M > N$. Desta forma, notei que, em todos os retângulos M por N , com $M > N > 0$, até a ordem 150 por 149, o ponto mais visitado é sempre o ponto $(1,0)$, sendo que, em algumas situações, ele não é o único com a quantidade máxima de caminhos. Este seria o ponto ideal para se vender os amendoins, portanto.

A compilação do programa rodou por uma noite inteira no Maxima. Caso eu aumentasse a ordem, certamente o programa iria demorar mais tempo. Os comandos e as respectivas saídas feitos no Maxima foram os seguintes:

```

--> c(x,y):=(x+y)!/x!/y!*(m-x+n-y)!/(m-x)!/(n-y)!;

(%o73) c(x,y):=
      (x+y)!
      x!
      y!
      (m-x+n-y)!
      (m-x)!
      (n-y)!

(%i5) mf:150;nf:149;
(mf)
150
(nf)
149

(%i6) for n:1 step 1 thru nf do
for m:n+1 step 1 thru mf do
for x:1 step 1/*ISTO É UM COMENTÁRIO ALEATÓRIO*/ thru m do
for y:0 step 1 thru entier(n*x/m) do
if ({x,y}# {m,n}) and (c(1,0)<c(x,y))
then
display("EPA!",m,n,[x,y]);
(%o6) done

```

Figura 3: Comandos e saídas no caso dos retângulos até ordem 150 por 149

Assim, como o Maxima não escreveu na tela a palavra “EPA” em nenhum dos retângulos considerados, concluímos que o ponto $(1,0)$ é o mais visitado para todos estes casos, de modo determinístico até a ordem 150 por 149. Tal constatação nos encoraja a fazer a seguinte

Conjectura: em qualquer retângulo M por N , $M > N$, o ponto mais visitado é, sempre, $(1,0)$ (não necessariamente o único com a quantidade máxima de caminhos).

Em uma aula de Estatística Básica ou de Combinatória no ensino superior o problema do ponto mais visitado pode ser, num primeiro momento, proposto aos alunos para que os mesmos possam tentar resolvê-lo à mão, para retângulos 3 por 2, 4 por 2, 4 por 3, etc. Em seguida, o professor pode instigá-los a conjecturar uma solução a partir da observação destes casos. Depois, o professor pode orientar os alunos a programarem no Maxima e, juntos, resolverem o problema até uma ordem escolhida, como fizemos acima. Enfim, o professor pode então mostrar aos estudantes que este problema encontra-se em aberto, que não há uma

prova definitiva para a questão, exceto para o quadrado, já solucionado por Santos e Castilho (2013), em cuja prova usou-se o Princípio de Indução, que também pode ser apresentada para os estudantes.

2.1 Simulações não determinísticas no software Python para os retângulos

No software Python também é possível conseguir boas razões para se formular a mesma conjectura do problema do ponto mais visitado em retângulos, porém, de forma não determinística. Como o Python não realiza simplificações algébricas, expressões envolvendo fatoriais de números grandes podem resultar em valores aproximados, assim, ao invés de se trabalhar com a fórmula de $c(x, y, M, N)$, optou-se por simular caminhos aleatórios saindo de um vértice e chegando em outro vértice do retângulo, por meio de milhares de repetições em vários tipos de retângulos de ordem $M > N$.

Para se sortear um caminho aleatório de um vértice a outro do retângulo, programei o Python para que executasse a seguinte tarefa: imaginando uma sacola da qual são retiradas ao acaso bolas brancas e azuis, onde cada bola branca representa um passo para a direita e cada bola azul para cima, o programa deve retirar bola por bola sem reposição, considerando a probabilidade de sair cada cor com base nas quantidades de bolas que ainda restam na sacola, até que se complete o percurso. O processo é repetido milhares de vezes, a fim de que seja possível identificar qual o ponto do plano mais visitado pelos milhares de percursos.

Foram realizadas duas simulações. A primeira em retângulos 10 por 4, por 1.000 vezes. Destas, 724 percursos passaram pelo ponto (1,0), sendo a maior quantidade registrada pelo programa. A segunda, em retângulo 8 por 3, por 2.000 vezes, das quais 1.480 passaram pelo ponto (1,0), sendo novamente a maior quantidade registrada.

Pode-se deduzir, a partir destes resultados não determinísticos, que o ponto (1,0) é o ponto mais visitado por percursos em retângulos M por N com $M > N$. O código programado no Python no qual foram realizadas as simulações está no apêndice I, ao final deste artigo. O software Python pode ser utilizado *online*, no site <https://repl.it/languages/python3>. O leitor pode, se assim o quiser, copiar os comandos descritos no apêndice deste artigo e colá-los no Python *online* para que possa executar as simulações.

3 Provas de dois casos particulares em retângulos

Apesar de não termos a demonstração do problema do ponto mais visitado para o caso genérico M por N em retângulos, temos alguns resultados de casos particulares. As proposições 1 e 2 a seguir são dois casos particulares que iremos demonstrar, e que também podem ser apresentados a estudantes de Matemática como forma de estímulo para a busca de uma prova para o caso geral, ainda em aberto:

Proposição 1: Para retângulos do tipo M por 1 ($M > 1$, $N = 1$), o ponto (1,0) é o mais visitado (não necessariamente o único com a quantidade máxima de caminhos).

Prova: Considere o reticulado com ponto final em $F = (k, 1)$, $k = M > 1$, como mostra a figura a seguir. Seja o ponto $P = (t, 0)$, $0 < t < k$. Afinal, quantos caminhos passam por P ? Primeiro, para chegar em P , só há um percurso, o que segue pela linha horizontal da origem até P . Agora, partindo de P e chegando em $F = (k, 1)$ há $k - t + 1$ caminhos, número que corresponde à quantidade de segmentos verticais de P a F , em destaque na figura:

Figura 4: Retângulos M por 1, $M > 1$

Logo, saindo da origem e chegando em F há $k - t + 1$ caminhos possíveis que passam por P .

De maneira análoga, pelo seu vizinho da direita $Q = (t + 1, 0)$, $0 < t + 1 \leq k$, passam $k - (t + 1) + 1$ caminhos. Então, por P passam mais caminhos do que por Q , e, assim, concluímos que, dentre os pontos de coordenadas inteiras, que estão sobre a reta $y = 0$, o ponto mais visitado é o ponto $(1, 0)$.

Por simetria, o ponto $(k - 1, 1)$ é o ponto mais visitado, em comparação agora com os pontos sobre a reta $y = 1$, tendo a mesma quantidade de caminhos que passam por $(1, 0)$. Enfim, está provada a proposição para o caso M por 1, $M > 1$.

Proposição 2: Para retângulos do tipo M por 2 ($M > 2$, $N = 2$), o ponto $(1, 0)$ é o mais visitado (não necessariamente o único com a quantidade máxima de caminhos).

Prova: Queremos provar que, para $(x, y) \neq (0, 0)$, $(x, y) \neq (M, 2)$, $0 \leq x \leq M$, $0 \leq y \leq 2$, vale:

$$c(x, y, M, 2) \leq c(1, 0, M, 2),$$

ou, equivalentemente,

$$\frac{(x + y)!}{x! y!} \cdot \frac{(M - x + 2 - y)!}{(M - x)! (2 - y)!} \leq \frac{M(M + 1)}{2} \quad (*).$$

Separaremos nos casos $y = 0, 1$ e 2 . Para $y = 0$, temos de provar:

$$\frac{(M - x + 2)(M - x + 1)}{2} \leq \frac{M(M + 1)}{2}.$$

Ora, como x não pode ser igual a zero nesta situação, já que por hipótese $(x, y) \neq (0, 0)$, então $M - x + 1 \leq M$ e $M - x + 2 \leq M + 1$, o que prova o caso $y = 0$.

Já para $y = 1$, temos de provar, a partir de (*):

$$\begin{aligned} (x + 1)(M - x + 1) &\leq \frac{M(M + 1)}{2} \Leftrightarrow \\ 2(Mx - x^2 + x + M - x + 1) &\leq M^2 + M \Leftrightarrow \\ M(2x + 1) + 2 &\leq M^2 + 2x^2 (**), \end{aligned}$$

$M > 2$ e $0 \leq x \leq M$.

Esta última desigualdade faremos por indução em M . Para $M = 3$, temos:

$$9 + 2x^2 \geq 6x + 5 \Leftrightarrow 2x^2 + 4 \geq 6x,$$

o que é verdade considerando cada possível valor de $x = 0, 1, 2, 3$.

Supondo a desigualdade (**) válida para $M \geq 3$ e $0 \leq x \leq M$, temos de provar válida também para $M + 1$, com $0 \leq x \leq M + 1$:

$$(M + 1)(2x + 1) + 2 \leq (M + 1)^2 + 2x^2 (**).$$

Vamos provar (**) começando “pelo final”, isto é, para $x = M + 1$:

$$(M + 1)[2(M + 1) + 1] + 2 \leq (M + 1)^2 + 2(M + 1)^2 \Leftrightarrow$$

$$M^2 + M \geq 2,$$

o que é verdade pois $M \geq 3$.

Agora, provemos (**), para $0 \leq x \leq M$:

$$(**) \Leftrightarrow$$

$$(M+1)(2x+1) + 2 \leq (M+1)^2 + 2x^2 \Leftrightarrow$$

$$M(2x+1) + (2x+1) + 2 \leq M^2 + 2x^2 + 2M + 1.$$

Ora, como $2x+1 \leq 2M+1$, resta provar

$$M(2x+1) + 2 \leq M^2 + 2x^2,$$

o que é verdade pela hipótese de indução (**), válida para $0 \leq x \leq M$, como queríamos demonstrar.

Para $y = 2$, temos que provar, por (*):

$$\frac{(x+2)(x+1)}{2} \leq \frac{M(M+1)}{2}.$$

Neste caso em que $y = 2$, x não pode ser igual a M , pois por hipótese $(x, y) \neq (M, 2)$. Assim, temos que $M > x$, logo, $M \geq x+1$ e $M+1 \geq x+2$, o que prova a desigualdade acima, concluindo, assim, a prova da proposição para os retângulos M por 2.

3.1 Observações sobre retângulos

Em se confirmando nossa conjectura de que o ponto $(1,0)$ é o mais visitado quando $M > N$, observamos que, para os casos em que $M = N+1 > N$, o ponto $(1,1)$ também é o mais visitado juntamente com $(1,0)$, pois, nesta situação:

$$c(1,0, N+1, N) = \frac{(2N)!}{(N!)^2},$$

e

$$c(1,1, N+1, N) = 2 \cdot \frac{(2N-1)!}{N!(N-1)!}.$$

Mas,

$$\frac{2(2N-1)!}{N!(N-1)!} = \frac{2N(2N-1)!}{N!^2} = \frac{(2N)!}{N!^2} = c(1,0, N+1, N).$$

Outra observação é que, em todos os retângulos, por simetria, $c(x, y, M, N) = c(M-x, N-y, M, N)$, logo, dizer que o ponto $(1,0)$ é o mais visitado para $M > N$, é o mesmo que dizer que $(M-1, N)$ é também o mais visitado, com a mesma quantidade máxima de caminhos.

Uma última observação é que, para os retângulos tais que $M < N$, o ponto mais visitado nos casos tratados aqui é o ponto $(0,1)$, juntamente com o seu correspondente simétrico $(M, N-1)$.

4 Percursos em paralelepípedos – abordagem numérica

Para se determinar o ponto mais visitado por percursos em paralelepípedos por meio de simulações no Maxima é necessário fazer uma adaptação nos comandos, acrescentando-se uma dimensão.

Realizando simulações em paralelepípedos, observei que, para o cubo de ordem 2 por 2 por 2, o ponto mais visitado por percursos é o ponto $(1,1,1)$, considerando novamente caminhos seguindo pelas direções positivas dos eixos. Para os demais cubos, o ponto mais visitado são os pontos $(1,0,0)$ ou $(0,1,0)$ ou $(0,0,1)$. Para paralelepípedos M por N por P , sendo M a maior coordenada, é o ponto $(1,0,0)$.

O código e a saída no Maxima das simulações nos paralelepípedos foram, até a ordem 15 por 15 por 15:

```
(%i10) c(x,y,z):=(x+y+z)!/x!/y!/z!*(m-x+n-y+p-z)!/(m-x)!/(n-y)!/(p-z)!;

$$c(x,y,z) = \frac{\frac{\frac{(x+y+z)!}{x!y!z!} \cdot (m-x+n-y+p-z)!}{(m-x)!} \cdot (n-y)!}{(p-z)!}$$

(%o10) c(x,y,z):=

$$\frac{(x+y+z)!}{x!y!z!} \cdot \frac{(m-x+n-y+p-z)!}{(m-x)!} \cdot \frac{(n-y)!}{(p-z)!}$$


(%i22) mf:15;nf:15;pf:15;
(mf) 15
(nf) 15
(pf) 15

--> for m:1 step 1 thru mf do
for n:1 step 1 thru nf do
for p:1 step 1 thru pf do
for x:0 step 1/*ISTO É UM COMENTÁRIO ALEATÓRIO*/ thru m do
for y:0 step 1 thru n do
for z:0 step 1 thru p do
if([x,y,z]#[0,0,0])and([x,y,z]#[m,n,p])and(c(0,0,1)<c(x,y,z))and(p>=m)and(p>=n)
then
display("EPA!",m,n,p,[x,y,z]);
EPA!=EPA!
m=2
n=2
p=2
[x,y,z]=[1,1,1]
(%o23) done
```

Figura 5: O código e a saída para as simulações em paralelepípedos

A figura acima revela que o ponto (1,1,1) é o mais visitado apenas no cubo 2 por 2 por 2. Nos demais paralelepípedos M por N por P , com P a maior coordenada, até a ordem 15 por 15 por 15, é o ponto (0,0,1).

Por consequência desta observação, os pontos mais visitados nos demais cubos são os pontos (1,0,0), (0,1,0) e (0,0,1), pela simetria.

5 Conclusão

Quando um problema não possui ainda uma solução analítica, o computador pode ser uma ferramenta de auxílio para ao menos se fazer alguma conjectura. Foi o que fizemos neste trabalho com relação ao problema de se determinar o ponto pelo qual passam mais percursos em retângulos e em paralelepípedos.

O problema do ponto mais visitado pode ser aplicado em aulas de análise combinatória a nível superior, conectando os alunos a problemas desafiadores e em aberto, instigando-os à pesquisa e à investigação científica da Matemática.

6 Referências

ANDRADE, Lenimar Nunes. Máxima: um completo programa de programação algébrica. **Revista do Professor de Matemática**, São Paulo, v. 30, n. 77, p. 36-40, 2012.

MELO, Antônio Luiz de.; SANTOS, Rogério César dos. Desigualdades no Triângulo de Pascal. **C.Q.D.- Revista Eletrônica Paulista de Matemática**, Bauru, v. 3, 75-84, 2014.

SANTOS, Rogério César dos. O problema do ponto mais visitado e a cadeia do viajante. **C.Q.D.- Revista Eletrônica Paulista de Matemática**, Bauru, v. 8, p. 19-26, 2016a.

SANTOS, Rogério César dos. Um estudo probabilístico sobre caminhos em reticulados quadrados. **C.Q.D.- Revista Eletrônica Paulista de Matemática**, Bauru, v. 8. p. 76-85, 2016b.

SANTOS, Rogério César dos; CASTILHO, José Eduardo. O problema do ponto mais visitado. **Revista do Professor de Matemática**, São Paulo, v. 31, n. 82, p. 50-53, 2013.

Apêndice I - código no software Python de simulação dos caminhos aleatórios em retângulos


```
print(" ")
import numpy as np
from random import *
# -*- coding: utf-8 -*-
n=int(input("Linhas:"))
m=int(input("Colunas:"))
print(" ")
mc=int(input("Quantas simulações quer fazer?"))
if (mc>20000):
    print(" ")
    print("Valor muito alto. Vou considerar apenas 20 mil simulações.")
    print(" ")
    input("Aperte enter:")
    mc=20000
if (mc<=0):
    print(" ")
    print("Inválido. Vou considerar uma única simulação.")
    print(" ")
    input("Aperte enter:")
    mc=1
mat2=np.zeros((n+1,m+1))
count = 1
while count <= mc:
    count += 1
    bolasb=m
    bolasp=n
    mat=np.zeros((n+1,m+1))

    # bola branca = 0 = para a direita
    # bola preta = 1 = para baixo
    mat2[0,0]+=1

    l=0
    c=0
    for k in range(n+m-1):
        if ((bolasb>0)and(bolasp>0)):
            t=np.random.choice([0,1],p=[bolasb/(bolasb+bolasp),bolasp/(bolasb+bolasp)])
            if (t==0):
                c=c+1
                mat2[l,c]+=1
                bolasb=bolasb-1
            else:
                l=l+1
                mat2[l,c]+=1
                bolasp=bolasp-1
        else:
            if(bolasb==0):
                l=l+1
```

```
    mat2[l,c]+=1
    bolasp=bolasp-1
    if(bolasp==0):
        c=c+1
        mat2[l,c]+=1
        bolasb=bolasb-1
    mat2[n,m]+=1
    if (count>mc):
        print(" ")
        print("RESULTADO FINAL DAS SIMULAÇÕES (quantos caminhos passaram por cada
ponto):")
        print(" ")
        print(mat2)
    if (n+m>40):
        print("Retângulo muito grande para encontrar o maior elemento da matriz.")
        print("Procurar visualmente.")
    else:
        print(" ")
        print("Aguarde enquanto eu encontro o maior dos elementos desta última matriz...")
#transformando a matriz em lista unidimensional:
    lista={}
    i=1
    for l in range (0,n+1):
        for c in range (0,m+1):
            if(((l>0)or(c>0))and((l<n)or(c<m))):
                i=i+1
                lista[i]=mat2[l,c]
#comparando os elementos da matriz2 com os da lista:
    nota=2
    for l in range (0,n+1):
        for c in range (0,m+1):
            if(not((l,c)==[0,0])or(l,c)==[n,m])):
                nota=0
                for k in range(2,(n+1)*(m+1)):
                    if (mat2[l,c]<lista[k]):

                        nota=1
    if(nota==0):
        linha=l
        coluna=c
    print(" ")
    input("Pronto! Aperte enter:")
    print(" ")
    print("O ponto mais visitado possui",mat2[linha,coluna],"caminhos:")
    print("Linha:",linha," e coluna",coluna)
```

Artigo recebido em ago. 2017 e aceito em nov. 2017.