

Cálculo de Áreas de Figuras Diversas

Aluna: Bianca Ortela André Novo
bi_ortel@yahoo.com.br

Orientadora: Maria Aparecida Diniz Ehrhardt
cheti@ime.unicamp.br

Instituto de Matemática, Estatística e Computação Científica (IMECC)

Agência financiadora: SAE/Unicamp
Palavras-chave: Áreas - Monte Carlo - Otimização

Introdução

O objetivo deste trabalho é estudar o **Método de Monte Carlo (MMC)** e sua aplicação no cálculo de áreas de figuras diversas, sobretudo áreas provenientes da intersecção ou união de outras. Esses casos são especiais, pois geralmente não temos uma expressão específica para calcularmos a área e, portanto, a utilização de uma simulação é uma idéia razoável para resolver esse problema. Além disso, trabalhamos com o problema de otimização relacionado ao cálculo das áreas. Nesse caso, o objetivo é minimizar a área resultante da intersecção de duas circunferências.

1 O MMC e o Cálculo de áreas

O MMC é uma técnica numérica que utiliza números aleatórios para resolver um problema. A primeira publicação do método talvez tenha sido o *paper* de Metropolis e Ulan *"The Monte Carlo Method"*, em 1949.

Uma das aplicações do MMC é o cálculo de áreas. Como este método utiliza números aleatórios, para o cálculo de áreas consideramos seqüências escolhidas ao acaso, independentes e uniformemente distribuídas no intervalo $[0, 1]$, que denotaremos como $U(0, 1)$. Além disso, se a seqüência x_n é $U(0, 1)$, então a transformação linear:

$$y_n = a + (b - a)x_n, \quad n = 1, 2, \dots,$$

onde a e b são números reais dados, se reduz à seqüência aleatória $y_n U(a, b)$.

Há diversas maneiras de se gerar números aleatórios e nesse caso optamos por utilizar a rotina *rand* do Matlab, que gera números aleatórios do tipo $U(0, 1)$.

1.1 Os problemas estudados

Escolhemos alguns problemas para aplicarmos o método de Monte Carlo e observarmos a eficiência do mesmo. Abaixo, temos a relação dos problemas escolhidos.

Figura 1: *Problemas estudados*

Problema / Intervalo	Restrições
1 - Área da circunferência de raio unitário / $[-1,1]$	$x_i^2 + y_i^2 \leq 1$
2 - Área intersecção de uma elipse e uma circunferência / $[0,1]$	$x_i^2 + y_i^2 \leq 1$ e $4x_i^2 + y_i^2 \geq 1$
3 - Área intersecção de duas elipses / $[0,1]$	$4x_i^2 + y_i^2 \leq 1$ e $x_i^2 + 4y_i^2 \leq 1$
4 - Área intersecção de uma circunferência e uma hipérbole / $[0,1]$	$x_i^2 + y_i^2 \leq 1$ e $4x_i^2 - 4y_i^2 \leq 1$
5 - Área intersecção de uma elipse e uma hipérbole / $[0,1]$	$x_i^2 + 4y_i^2 \leq 1$ e $4x_i^2 - 4y_i^2 \geq 1$
6 - Área intersecção de duas circunferências / $[0,1]$	$x_i^2 + y_i^2 \leq 1$ e $(x_i - 1)^2 + (y_i - 1)^2 \leq 1$
7 - Área intersecção de uma circunferência e uma reta / $[0,1]$	$x_i^2 + y_i^2 \leq 1$ e $x_i + y_i \leq 1$
8 - $\frac{1}{4}$ de área de uma estrela de 4 pontas / $[0,1]$	$x_i - 2y_i \geq -1$ e $2x_i - y_i \leq 1$
9 - Área da estrela de David / $[-1,1]$	$1,73x + y \leq 1$ e $-1,73x + y \leq 1$ e $y \geq -0,5$ ou $1,73x - y \leq 1$ e $-1,73x - y \leq 1$ e $y \geq 0,5$

Para acharmos a área de cada um desses problemas, basta gerarmos N pares de números aleatórios independentes, uniformemente distribuídos no intervalo relativo a cada um dos problemas. Com os pontos em mãos, devemos verificar quais satisfazem as restrições. Assim, computamos n , o número de pontos para os quais as restrições são satisfeitas. Dessa forma, o cálculo da área é dada por:

$$A = A_{quadrado} \frac{n}{N} u^2,$$

onde n é o número de pontos interiores à figura considerada, N o número de pontos total, a $A_{quadrado}$ é relativa ao intervalo considerado e u^2 indica a unidade de área arbitrária.

1.2 Resultados Obtidos

Para o cálculo das áreas, foi implementado um programa que segue o procedimento descrito acima. O programa foi rodado 20 vezes e obtivemos uma média para o valor de cada uma das áreas e seu respectivo desvio padrão. Nota-se que à medida que N cresce, o valor simulado para a área se torna mais próximo do valor real.

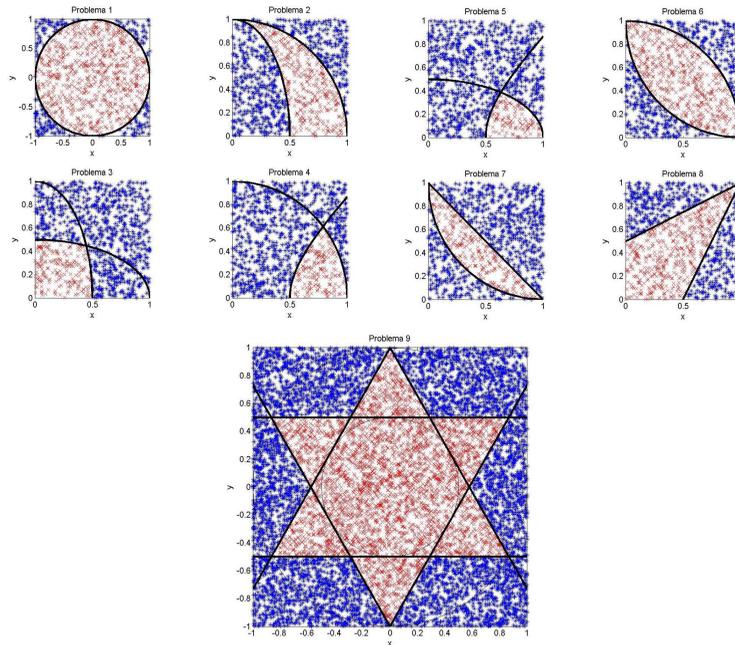
Figura 2: *Resultados Numéricos*

Nº de Pontos	100		1000		10000		100000		Valor
	Média	Desvio	Média	Desvio	Média	Desvio	Média	Desvio	
1	3,1670	0,2359	3,1148	0,0563	3,1407	0,0676	3,1446	0,0182	3,1415
2	0,3890	0,0427	0,3908	0,0170	0,3934	0,0046	0,3933	0,0017	0,3927
3	0,2435	0,0382	0,2330	0,0190	0,2293	0,0060	0,2320	0,0014	0,2318
4	0,2095	0,0539	0,1976	0,0101	0,1995	0,0038	0,2004	0,0011	0,2005
5	0,1375	0,0286	0,1335	0,0085	0,1316	0,0030	0,1320	0,0009	0,1324
6	0,5640	0,0562	0,5716	0,0141	0,5711	0,0033	0,5713	0,0019	0,5707
7	0,2935	0,0534	0,2853	0,0116	0,24844	0,0044	0,2852	0,0017	0,2854
8	0,4960	0,0469	0,5024	0,0172	0,4997	0,0061	0,4994	0,0020	0,5
9	1,7020	0,2212	1,7093	0,1352	1,7404	0,0220	1,7363	0,0050	1,732

A seguir, estão as figuras representativas de uma simulação de cada problema levantado, com $N = 1000$, nos problemas de 1 a 8 e $N = 10000$, no problema 9.

Os pontos vermelhos são os interiores à figura desejada e os azuis são externos à mesma.

Figura 3: *Problemas de 1 a 9*



2 O problema de otimização associado

A idéia agora é, dados pontos escolhidos, encontrar os parâmetros das figuras que nos fornecem a menor área que os contenha, ou seja, encontrar curvas ótimas.

Escolhemos um conjunto pequeno de pontos. Para cada conjunto de pontos, temos um conjunto de parâmetros associado, que deverá nos fornecer a menor área possível. Nesse caso, a função objetivo do nosso problema será a área intersecção ou união das figuras e as restrições serão os pontos escolhidos estarem no interior ou na fronteira das curvas. No caso da intersecção de duas circunferências temos o seguinte problema, onde (y_i, z_i) são os pontos escolhidos.

$$\begin{aligned} & \min f(x) \\ \text{su}j. & a (y_i - x_1)^2 + (z_i - x_2)^2 \leq x_3 \\ & (y_i - x_4)^2 + (z_i - x_5)^2 \leq x_6, \quad x \in R^6. \end{aligned} \quad (1)$$

Como $f(x)$ não possui uma fórmula explícita, contornaremos esse problema usando a simulação de Monte Carlo. A cada iteração, calculamos a área intersecção das figuras envolvidas no problema, isto é, das duas circunferências, e esse será o valor de $f(x)$. Optamos por resolver esse problema usando o método do Lagrangiano aumentado, uma vez que a tentativa de utilizar o método de penalização externa fracassou. Assim, nosso problema pode ser identificado como:

$$\begin{aligned} & \min f(x) \\ \text{su}j. & a h_i(x) \leq 0, \quad i = 1, \dots, t. \end{aligned} \quad (2)$$

Para esse problema com restrições de desigualdade, Rockafellar,[4], sugere a seguinte generalização da função Lagrangiano aumentado:

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^m Q_i(x), \quad (3)$$

onde $Q_i(x)$ é definida como:

$$\text{Se } h(x) \geq \frac{-\lambda_i}{\mu}, \text{ então } Q_i(x) = \lambda_i h_i(x) + \mu * h_i(x)^2;$$

$$\text{Se } h(x) < \frac{-\lambda_i}{\mu}, \text{ então } Q_i(x) = \frac{-\lambda_i^2}{4\mu}.$$

Com isso, aplicamos o algoritmo do Lagrangiano aumentado para restrições de desigualdade e no passo minimizar $L(x, \lambda, \mu)$, utilizamos a rotina *fminsearch* do Matlab. Porém, não conseguimos encontrar uma solução ótima para o problema, em virtude do fato de a rotina *fminsearch* necessitar de grande número de avaliações da função. Encontramos apenas uma solução factível. Além disso, concluímos que, com um tempo de computação razoável, não conseguiríamos uma solução ótima e, portanto, nos contentamos com uma solução factível. Dessa forma, fixamos o conjunto de pontos chutados em 5; fixamos o número máximo de avaliações da função em 5000 e o número máximo de iterações em 2500.

2.1 Resultado do experimento

Dados iniciais: $x_0 = [0 \ 0 \ 1 \ 1 \ 1 \ 1]$; $\lambda = [0 \ 0 \ 0 \ 0 \ 0 \ 0]$; $\mu = 1$; $erro = 10^{-4}$.
Pontos escolhidos: $(y, z) = [(1, 0), (0, 1), (0.5, 0.5), (0.5, 0.34), (0.3, 0.4)]$.

Com esses dados, após a aplicação do algoritmo do Lagrangiano aumentado para restrições de desigualdade, obtivemos os seguintes resultados.

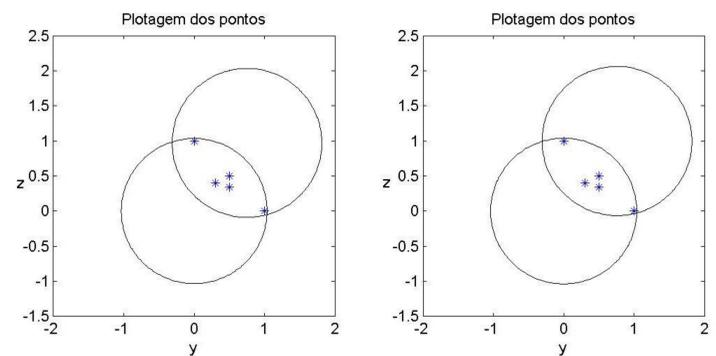
Figura 4: *Resultados*

Solução	Vetor de parâmetros	Área
Sem refinamento	$xmin = [0.0005 \ 0.0003 \ 1.0728 \ 0.7516 \ 0.9734 \ 1.1305]$	1.0343 u^2
Com refinamento	$xmin = [0.0005 \ 0.0003 \ 1.0806 \ 0.7580 \ 1.0004 \ 1.1342]$	1.0022 u^2

A área foi calculada posteriormente, com base nos parâmetros das curvas fornecidos por *xmin* e da forma como fizemos na etapa anterior. Calculamos a área 100 vezes e tomamos o valor médio. O desvio padrão nesse caso foi de 0.0165 u^2 , para o caso sem refinamento e 0.0173 u^2 para o caso com refinamento.

Verificamos que a diferença entre os valores obtidos para área é da ordem de 10^{-2} e, portanto, o refinamento da solução parece ter surtido efeito. Entretanto, este refinamento não pode ser feito ilimitadamente, porque a rotina, logo após o primeiro deles, não avança para soluções melhores. Abaixo, estão as figuras relacionadas às soluções:

Figura 5: *Solução com refinamento / sem refinamento*



Nas figuras não se pode verificar muita diferença entre a solução com refinamento e a solução sem refinamento, até porque os parâmetros das curvas são muito parecidos. Apenas com o cálculo da área verificamos a diferença, que também é pequena.

3 Conclusões

O MMC se mostrou eficiente por ser de fácil implementação, por ser rápido e não exigir muito esforço da máquina e por fornecer valores para as áreas muito próximos dos valores reais. Assim, verificamos que poderíamos aplicá-lo para resolver o problema de minimizar a área intersecção ou união de figuras geométricas quaisquer. E para isso, utilizamos o método do Lagrangiano aumentado associado ao MMC. Dessa forma, não conseguimos encontrar uma solução ótima para o problema em virtude da estrutura do mesmo e da ferramenta computacional utilizada. Porém, conseguimos uma solução factível e a aplicação do refinamento se mostrou eficiente, o que indica que a abordagem do problema da forma como fizemos estava na direção correta.

4 Referências

- [1] Murthy, K.P.N., *An Introduction to Monte Carlo Simulations in Statistical Physics*, Theoretical Studies Section - Materials Science Division, Indira Gandhi Center for Atomic Research, India, 16 de dezembro de 2003.
- [2] Luenberger, D.G., *Linear and Nonlinear Programming*, 2ª edição, Addison - Wesley Publishing Company, Nova York, 1986.
- [3] J.M.Martinez & S. A. Santos. *Métodos Computacionais de Otimização - XX Colóquio Brasileiro de Matemática*. IMPA, 1995.
- [4] R.T. Rockafellar. Augmented Lagrange multiplier functions and duality in non-convex programming. *SIAM J. Control and Optimization* 12, 1974.

