O algorítmo de Passeio Aleatório (PA) é um método de Monte Carlo baseado nas Cadeias de Markov (MCCM) da classe de algorítmos de *Metropolis-Hastings* (MH), diferenciando-se em $x_t \sim \mathcal{N}(x_{t-1}, \sigma^2)$, o que leva a uma simetria da distribuição de probabilidade (d.p.).

O algorítmo de *Metropolis-Hastings* consiste no seguinte:

1. Escolher um x_0 aleatório do domínio da função de d.p. objetivo $f(x) \propto P(x)$, e escolher uma d.p. candidata G (com função de d.p. $g(x_t \mid x_{t-1})$).
   O PA usará a distribuição normal $G = \mathcal{N}(x_{t-1}, \sigma^2)$ como candidata *a priori*. Será preciso também escolher σ, o que representará o desvio padrão da distribuição.
2. Gerar um x_t^* candidato da d.p. candidata
3. Gerar u a partir de $\mathcal{U}(0, 1)$ (Monte Carlo)
4. Calcular $\alpha = \frac{f(x_t^*)/g(x_t^*|x_{t-1})}{f(x_{t-1})/g(x_{t-1}|x_t^*)} = \frac{f(x_t^*)}{f(x_{t-1})} \frac{g(x_{t-1}|x_t^*)}{g(x_t^*|x_{t-1})}$
   Como a distribuição normal é simétrica, $\frac{g(x_{t-1}|x_t^*)}{g(x_t^*|x_{t-1})} = 1$, logo $\alpha = \frac{f(x_t^*)}{f(x_{t-1})}$
5. Comparar u com α
   Se $\alpha > u$ então x_t^* é aceite como x_t^*, e o algoritmo repete a partir do ponto 2 com o novo $t \leftarrow t + 1$ (ou acaba dependendo da quantidade da amostra que pretendemos)
   Se não, então x_t^* é rejeitado e o algorítmo volta para o ponto 2 sem alterar t

Como isto é uma cadeia de Markov, para uma amostra suficientemente grande, há uma convergência para a distribuição $f(x)$.

Os gráficos ilustrados apresentam a importância da escolha do desvio padrão σ.

AGR É TEU

Os gráficos ilustrados apresentam a convergência para a distribuição Qui-Quadrado com 2 graus de liberdade, sendo a principal diferença o desvio padrão e, por conseguinte, a variância.

No gráfico 1, o desvio padrão da distribuição proponente simétrica é 0.05, fazendo com que gere somente valores consecutivos altos, em que todos são aceites. No gráfico 2, à semelhança do 3, vemos que a função converge para o domínio da Qui-Quadrado, aceitando quase na sua totalidade os pontos gerados. Contudo, para o gráfico 2 a convergência é lenta e no gráfico 3 é bastante mais elevada, devido à variância ser maior (diferença dos passos seguintes ser maior). No gráfico 4, em contradição com o gráfico 1, a convergência é efetuada muito rapidamente, contribuindo para que o algoritmo não aceite pontos gerados e fique preso no mesmo sítio na execução das iterações (ter retas horizontais nas iterações vendo que permaneceu no último estado).

Talvez o mais ideal seria optar pelo gráfico 3 ou por um valor de desvio padrão entre os dois últimos gráficos, de forma a melhorar a eficácia da convergência, mas não rejeitando os pontos tal como acontece no gráfico 4, para um desvio padrão de 16.