# Algoritmos de Otimização para Construção de Tesselações Centroidais de Voronoi

Arthur Gabriel de Santana, Orientador: Ernesto G. Birgin

*Trabalho de Formatura Supervisionado*

26 de Novembro de 2018

Instituto de Matemática e Estatística – Universidade de São Paulo

## Diagrama de Voronoi

Motivação:

- $m$ caixas eletrônicos
- cada pessoa vai sempre ao caixa mais próximo

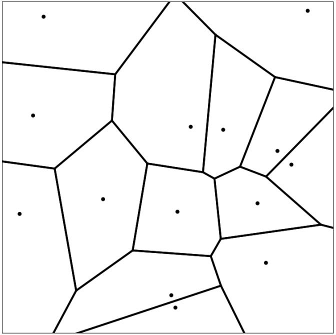
*Problema: como determinar, eficientemente, a região servida por cada caixa eletrônico?*

# Diagrama de Voronoi

Motivação:

- $m$ caixas eletrônicos
- cada pessoa vai sempre ao caixa mais próximo

*Problema: como determinar, eficientemente, a região servida por cada caixa eletrônico?*

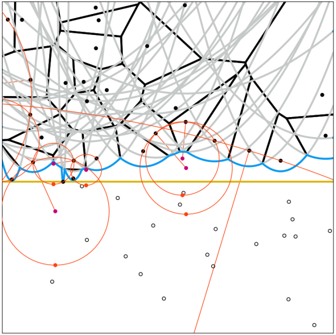


Queremos construir $V(Z) = \{V_i(Z)\}_{i=1}^{m}$, com

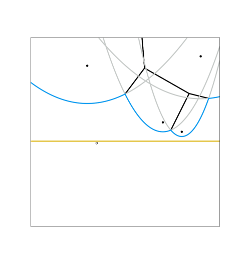
$$V_i(Z) = \{x \in \Omega \mid \|x - z_i\| < \|x - z_j\|, \forall j \neq i\}.$$

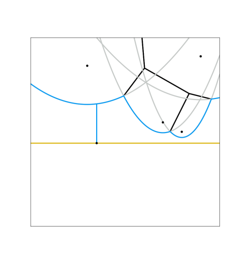
# Algoritmo de Fortune

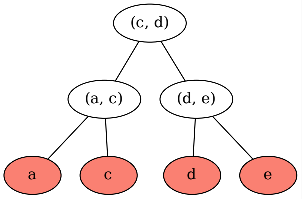
- Algoritmo de linha de varredura
- Linha da praia
- Eventos ponto: arestas do diagrama
- Eventos círculo: vértices do diagrama

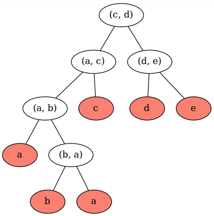


# Algoritmo de Fortune – ABB

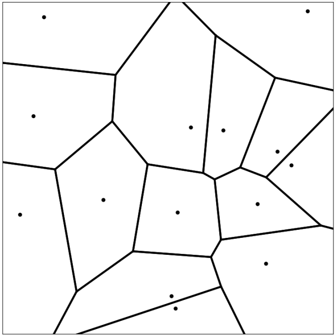






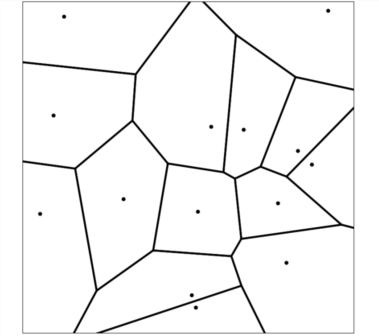



# Tesselação Centroidal de Voronoi



*Problema: como escolher o melhor lugar para os caixas eletrônicos?*

# Tesselação Centroidal de Voronoi



*Problema: como escolher o melhor lugar para os caixas eletrônicos?*

Minimizando a função

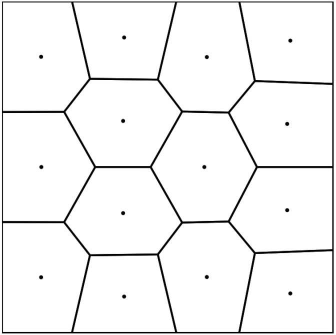
$$\mathcal{G}(Z) = \sum_{i=1}^{m} \int_{V_i(Z)} \|x - z_i\|^2 \rho(x) \, \mathrm{d}x \, .$$

# Tesselação Centroidal de Voronoi

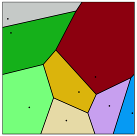
TCV: Diagrama de Voronoi em que os pontos geradores coincidem com os centros de massa.

$$C_i(Z) = \frac{\int_{V_i(Z)} x\rho(x)\,\mathrm{d}x}{\int_{V_i(Z)} \rho(x)\,\mathrm{d}x}$$

*Problema: como construir?*



# Algoritmo de Lloyd

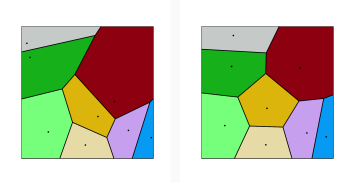


- método iterativo
- baseado na transformação

$$T(Z) = \{C_i(Z)\}_{i=1}^m.$$

- pontos fixos de $T$ são TCVs

# Algoritmo de Lloyd



- método iterativo
- baseado na transformação

$$T(Z) = \{C_i(Z)\}_{i=1}^{m}.$$

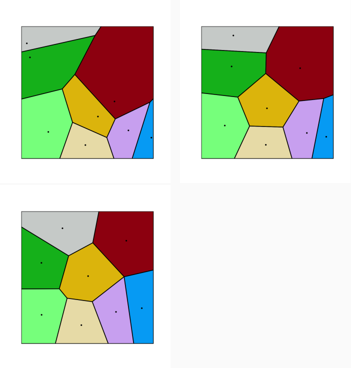
- pontos fixos de $T$ são TCVs

# Algoritmo de Lloyd

- método iterativo
- baseado na transformação

$$T(Z) = \{C_i(Z)\}_{i=1}^{m}.$$

- pontos fixos de $T$ são TCVs

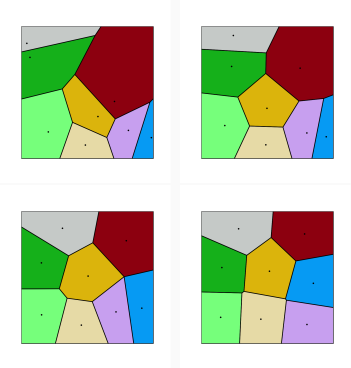


# Algoritmo de Lloyd

- método iterativo
- baseado na transformação

$$T(Z) = \{C_i(Z)\}_{i=1}^{m}.$$

- pontos fixos de $T$ são TCVs

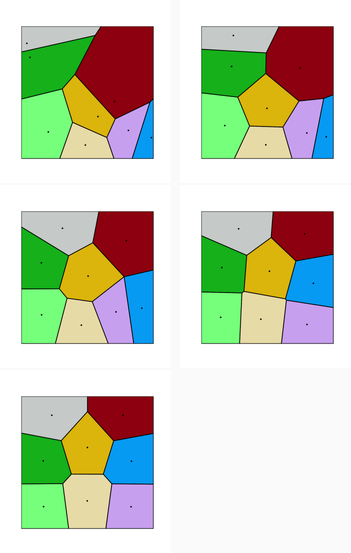


# Algoritmo de Lloyd

- método iterativo
- baseado na transformação

$$T(Z) = \{C_i(Z)\}_{i=1}^{m}.$$

- pontos fixos de $T$ são TCVs

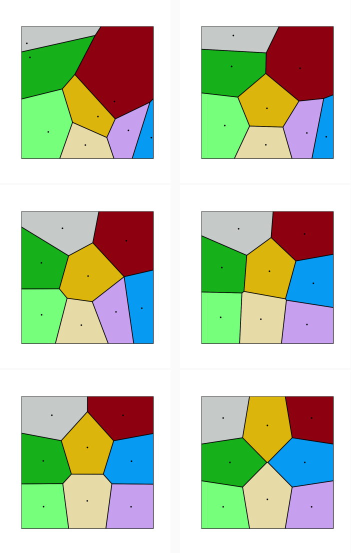


# Algoritmo de Lloyd

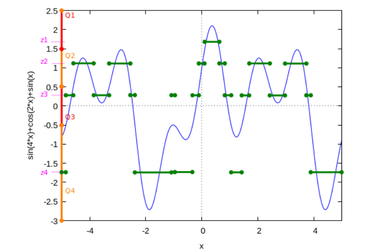
- método iterativo
- baseado na transformação

$$T(Z) = \{C_i(Z)\}_{i=1}^{m}.$$

- pontos fixos de $T$ são TCVs



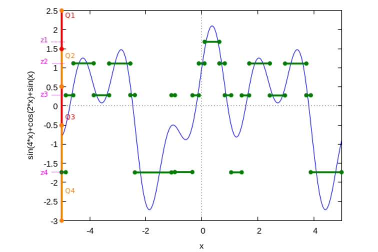
# Algoritmo de Lloyd – Quantização Ótima




- sinal original $s(t)$
- sinal quantizado $r(t)$
- distribuição $\rho$ dos valores
- ruído $n(t) = s(t) - r(t)$

*Queremos minimizar* $E[n^2]$

# Algoritmo de Lloyd – Quantização Ótima




- sinal original $s(t)$
- sinal quantizado $r(t)$
- distribuição $\rho$ dos valores
- ruído $n(t) = s(t) - r(t)$

*Queremos minimizar* $E[n^2] =$

$$\sum_{i=1}^{m} \int_{Q_i} n(t)^2 \, \rho(t) \, \mathrm{d}t \ .$$

# Método do Gradiente

$$\nabla \mathcal{G}(Z) = 2M(Z - T(Z))$$

- $d = -\nabla \mathcal{G}(Z)$
- $x^{k+1} = x^k + \lambda d$
- *backtracking*

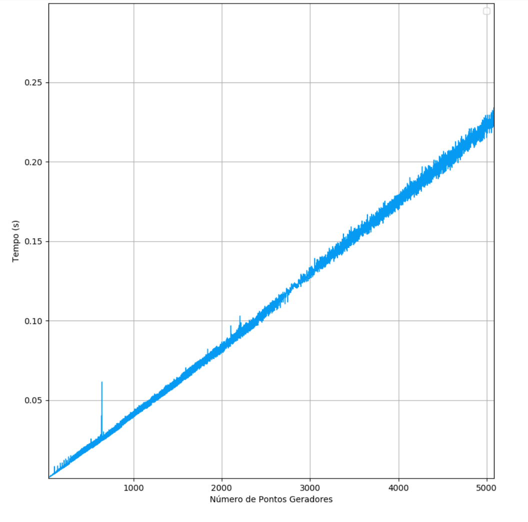
# Método do Gradiente

$\nabla \mathcal{G}(Z) = 2M(Z - T(Z))$

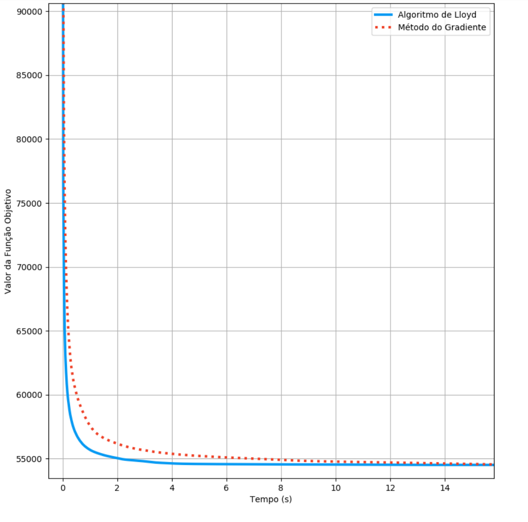
- $d = -\nabla \mathcal{G}(Z)$
- $x^{k+1} = x^k + \lambda d$
- *backtracking*

A partir da mesma ideia podemos utilizar outros métodos clássicos, com convergência mais rápida (Newton, L-BFGS, etc)

# Testes – Fortune

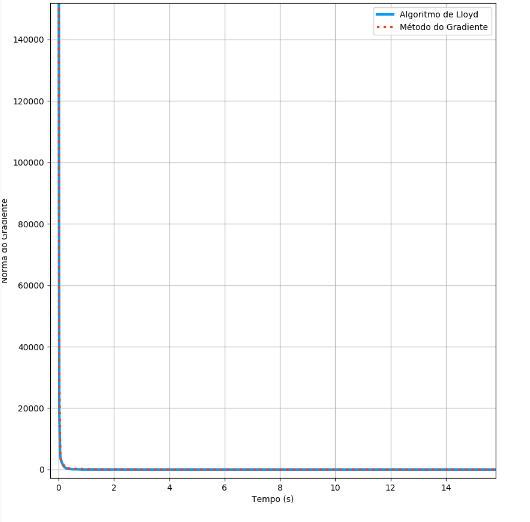


# Testes – TCVs



Algoritmo de Lloyd
Método do Gradiente
Valor da Função Objetivo
Tempo (s)
90000
85000
80000
75000
70000
65000
60000
55000
0
2
4
6
8
10
12
14

# Testes – TCVs



Algoritmo de Lloyd
Método do Gradiente
Norma do Gradiente
Tempo (s)
140000
120000
100000
80000
60000
40000
20000
0
0
2
4
6
8
10
12
14

# Próximos Passos

- estender a função objetivo com critérios de qualidade
- utilizar métodos mais avançados de otimização

# Referências


- BERG, M. et al. Computational Geometry. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2008.
- DU, Q.; FABER, V.; GUNZBURGER, M. Centroidal Voronoi Tessellations: Applications and Algorithms. SIAM Review, v. 41, n. 4, p. 637-676, 1999.
- FORTUNE, S. A sweepline algorithm for Voronoi diagrams. Algorithmica, v. 2, n. 1-4, p. 153-174, 1987.
- FRIEDLANDER, A. Elementos de Programação Não-Linear, Editora da Unicamp, 1994.
- LLOYD, S. Least squares quantization in PCM. IEEE Transactions on Information Theory, v. 28, n. 2, 1982.