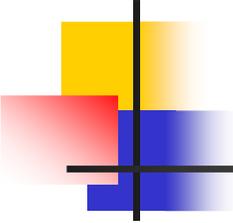


Geometria Computacional Aplicada

Márcio de Castro Marques
Raphael Silva Cury

Maio/2005

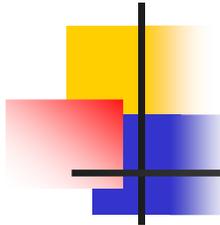


Sumário

- Redes de Sensores sem Fio (RSSF)
- Geometria Computacional
- Geometria Computacional e RSSF
- Aplicações
- Conclusões



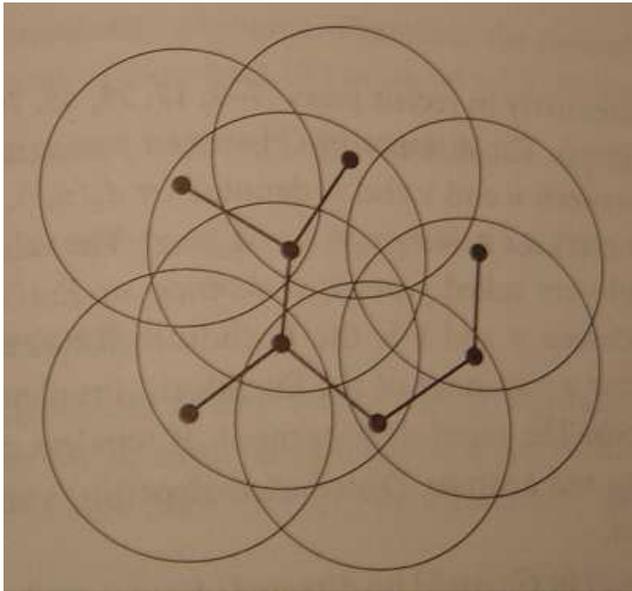
- Conjunto de nós sensores espalhados em uma área geográfica
 - Capacidade de processamento
 - Capacidade de sensoriamento
 - Capacidade de comunicação
- Grande número de nós sensores e dinamicidade do ambiente
 - Desafios no desenho de RSSF



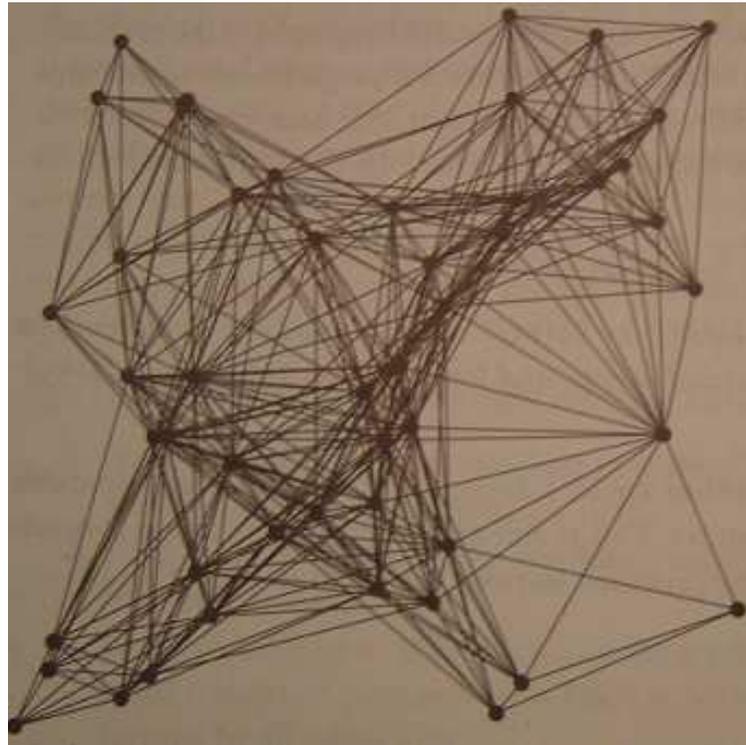
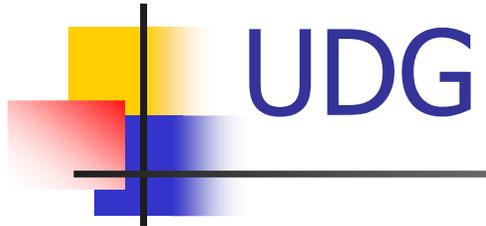
RSSF

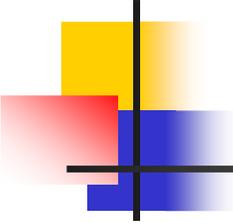
- RSSF pode ser vista como um grafo:
 - Nós sensores representam os nós do grafo
 - Cada sensor tem uma antena omnidirecional
 - Transmissão em todas as direções, pode ser considerada um disco centrado no nodo.
 - Raio do disco chamado de alcance de transmissão
 - Ligações entre o nó sensor e os nós sensores dentro do alcance de transmissão: arestas do grafo
 - Grafo de disco de unidade (Unit Disk Graph – UDG)

Geometria Computacional - UDG



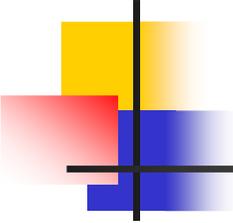
- RSSF constituída de um conjunto V de nós sensores
- Alcance de transmissão máximo de uma unidade
- Define um Grafo de Disco de Unidade em que um vértice existe entre dois nós apenas se sua distância euclidiana for no máximo de uma unidade





Spanners

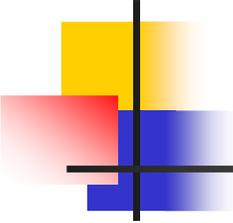
- Menor distância entre dois vértices u e v em um grafo $G(V,E)$: $d_G(u,v)$
- Sub-grafo H é T -Spanner de G se, para todo vértice u e v , $d_H(u,v) \leq t \cdot d_G(u,v)$
 - Valor de t chamado de "stretch value" (valor de estiramento)



Power stretch factor

- Menor caminho entre os vértices u e v em G
 - A energia consumida para percorrer este caminho é $p_G(u,v)$
- Menor caminho entre os vértices u e v no grafo H (sub-grafo de G)
 - A energia consumida para percorrer este caminho é $p_H(u,v)$
- Power Stretch Factor de H em relação a G :

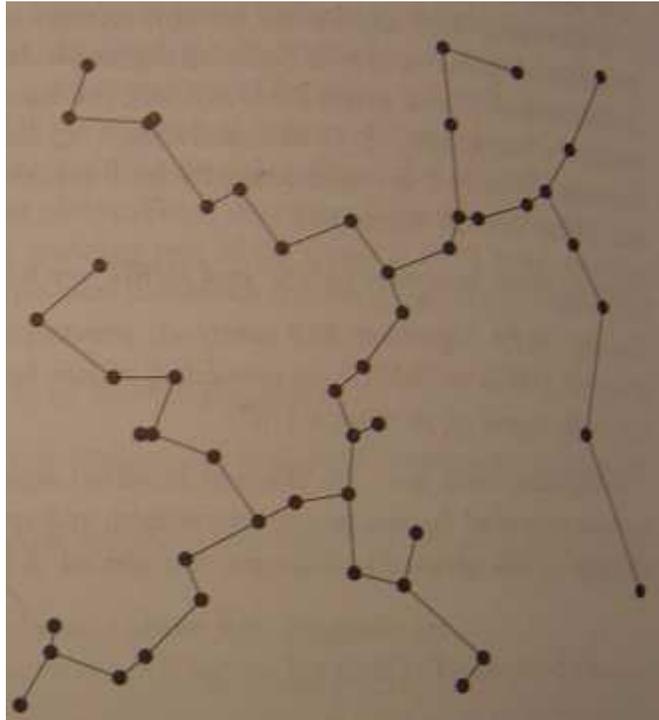
$$\rho_H(G) = \max [p_H(u,v) / p_G(u,v)]$$



Estruturas Geométricas

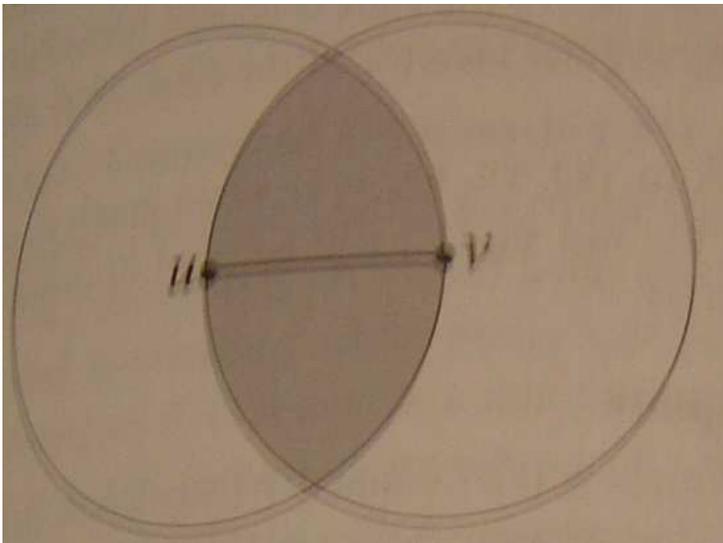
- **Árvore Geradora Mínima**
 - MST(G)
 - Árvore que conecta todos os nós e que o comprimento total dos vértices é minimizado
 - Um dos sub-grafos conexos mais esparsos
 - Fator de estiramento pode ser tão grande quanto $n-1$

Estruturas Geométricas - MST(G)



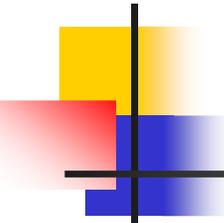
Estruturas Geométricas

- Grafo da vizinhança relativa

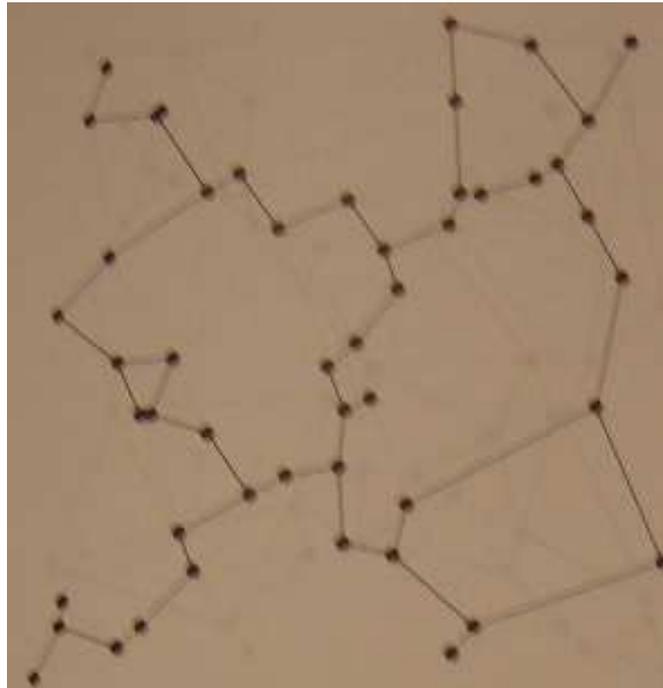


- $RNG(G)$

- Consiste de todas as arestas uv , tal que não exista um vértice w em V com arestas uw e wv em E satisfazendo $uw < uv$ e $wv < uv$

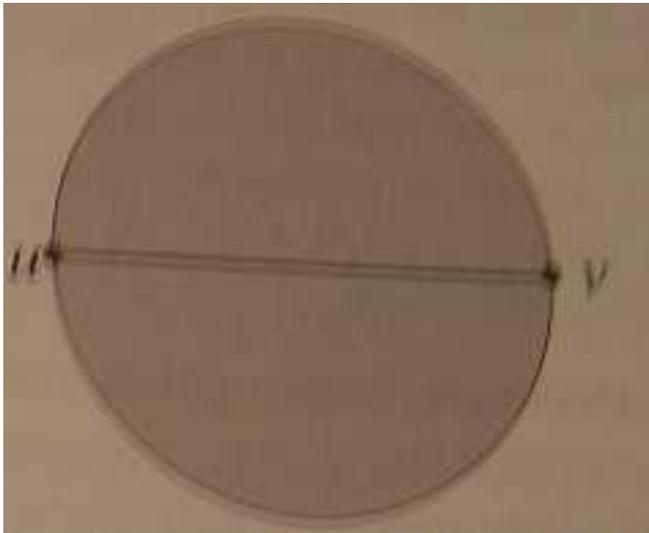


Estruturas Geométricas – RNG(G)

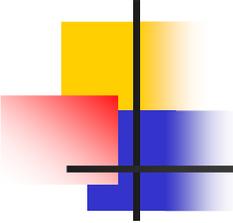


Estruturas Geométricas

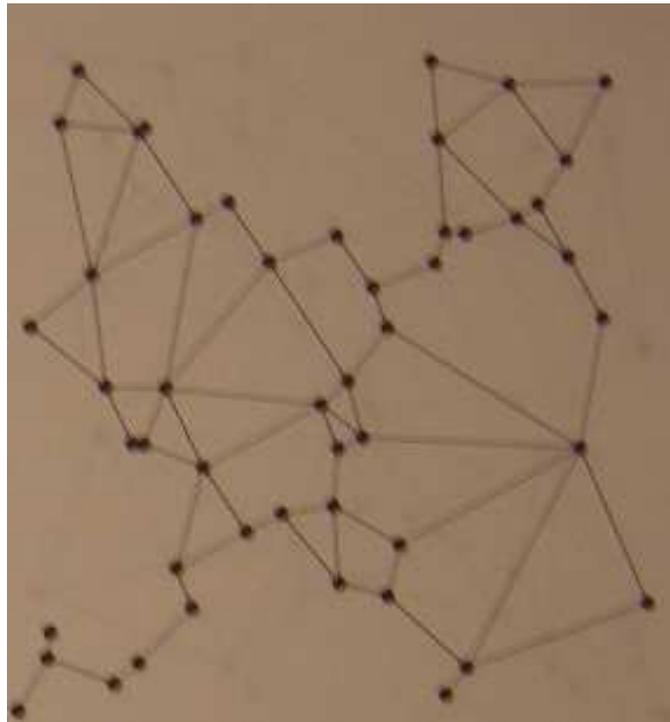
- Grafo de Gabriel



- $\text{GG}(G)$
- $\text{Disco}(u,v)$ é o disco com diâmetro uv
- Grafo de Gabriel contém a aresta uv apenas se $\text{Disco}(u,v)$ não contém outro vértice w de tal forma que existam arestas uw e wv

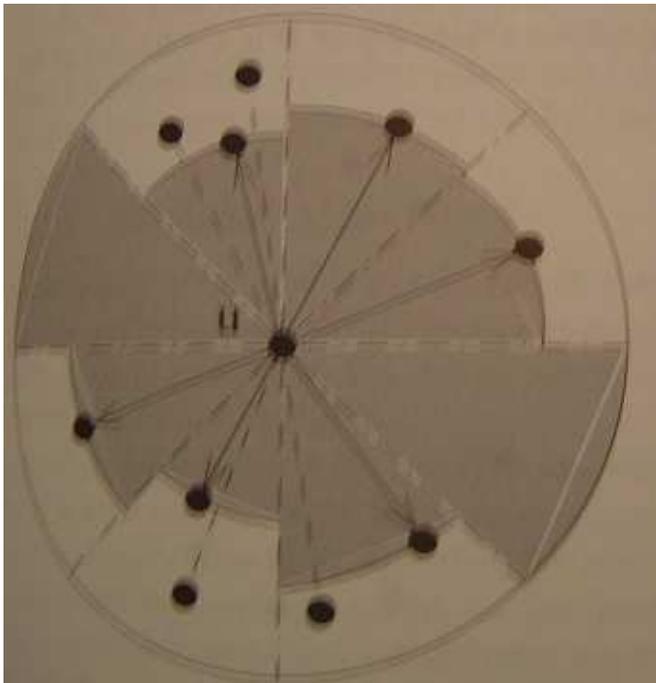


Estruturas Geométricas – GG(G)



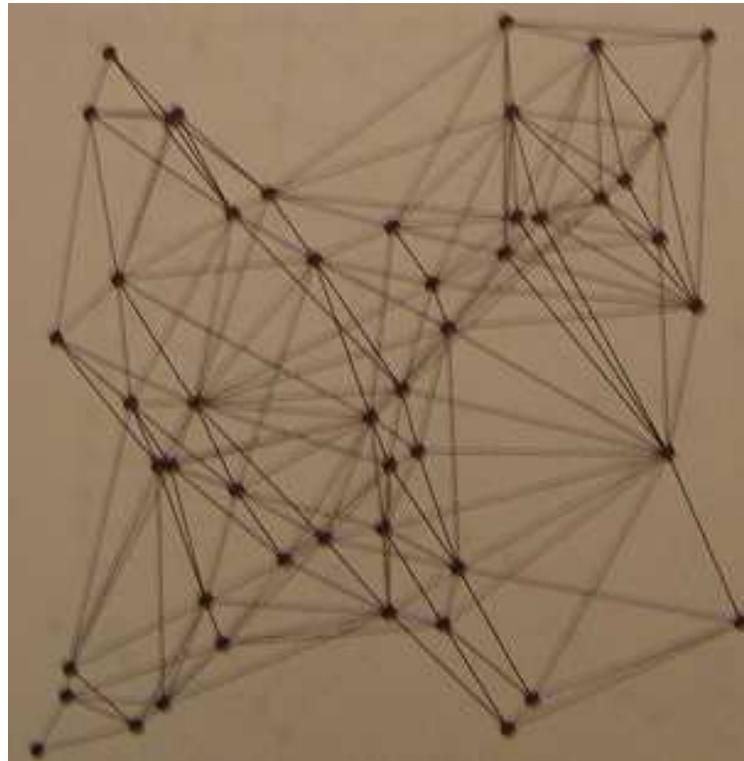
Estruturas Geométricas

■ Grafo de Yao

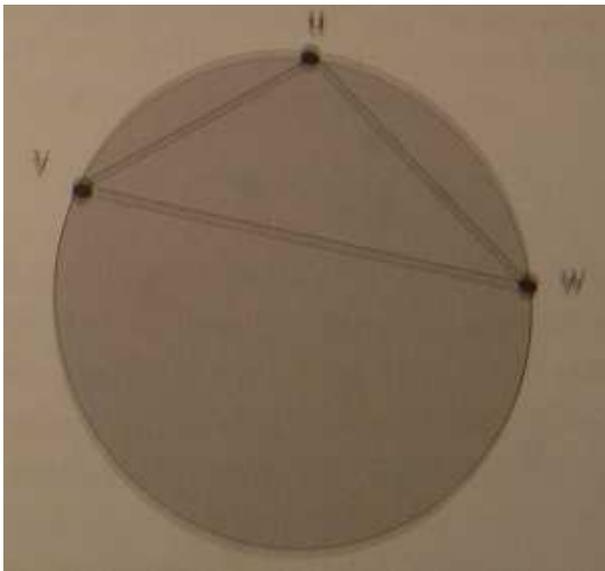


- $YG_k(G)$
- Para cada vértice u , k raios igualmente separados definem k cones ($K \geq 6$)
- Em cada cone, escolha o menor vértice uv entre todos os vértices de u
- Grafo θ : Escolhe o vértice de menor projeção no eixo do cone

Estruturas Geométricas – $YG_k(G)$

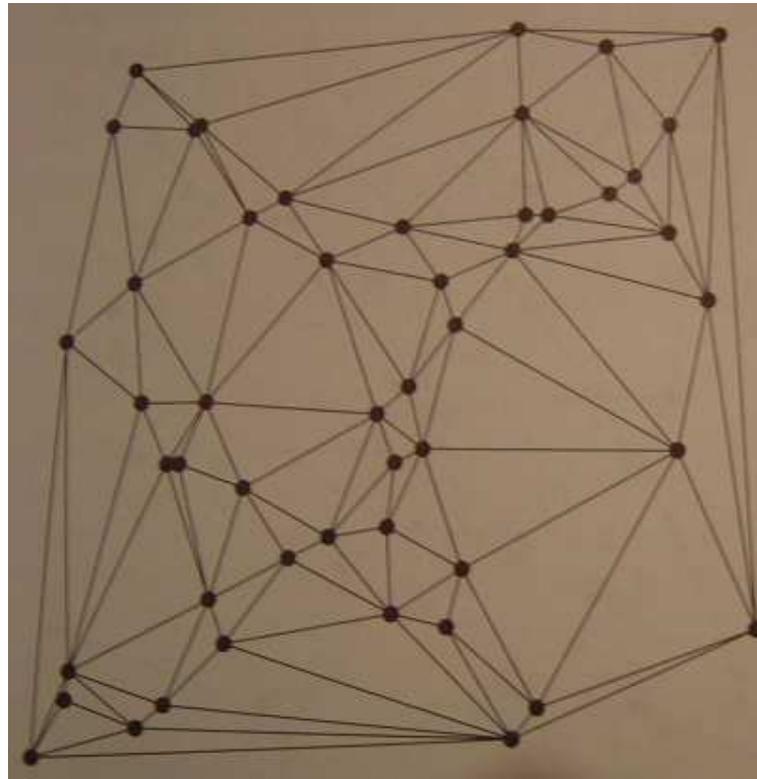
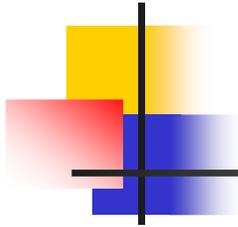


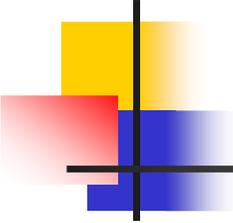
Estruturas Geométricas



- Triangulação de Delaunay
 - $\text{Del}(V)$
 - Assume que nenhum 4 vértices são co-circulares
 - Triangulação é de Delaunay se o círculo circunscrito de cada um de seus triângulos não contém outro vértice em seu interior

Estruturas Geométricas - Delaunay



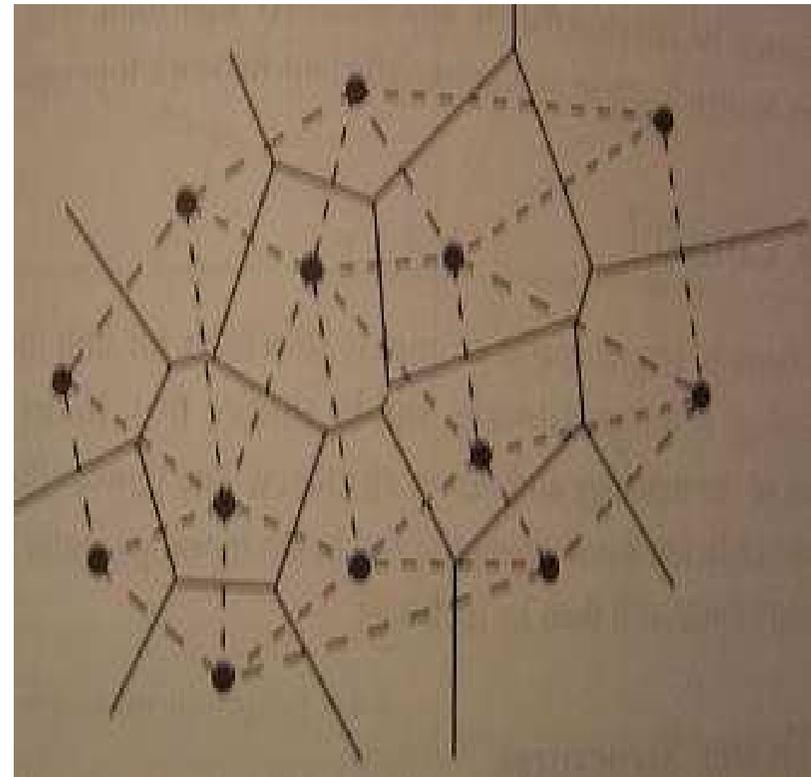


Estruturas Geométricas

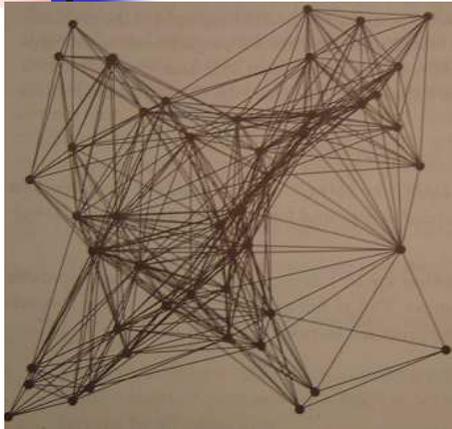
- Região de Voronoi
 - $Vor(p)$
 - Coleção de pontos bidimensionais, onde todos os pontos estão mais perto do vértice p do que de qualquer outro vértice no grafo
 - Diagrama de Voronoi é a união de todas as regiões de Voronoi do grafo

Estruturas Geométricas

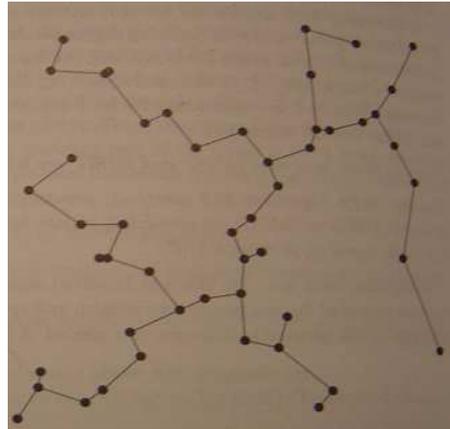
- Triangulação de Delaunay está relacionada com Diagrama de Voronoi
 - Dois vértices p e q estão conectados em $\text{Del}(V)$ apenas se $\text{Vor}(p)$ e $\text{Vor}(q)$ compartilham uma mesma borda



Estruturas Geométricas



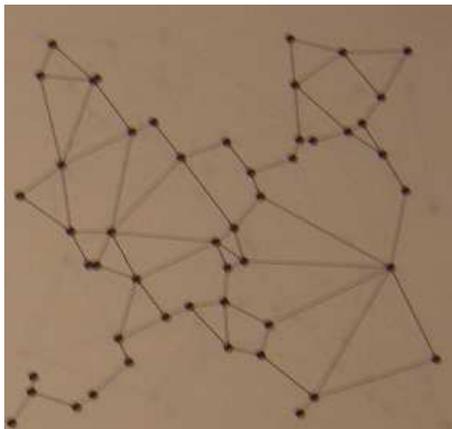
UDG



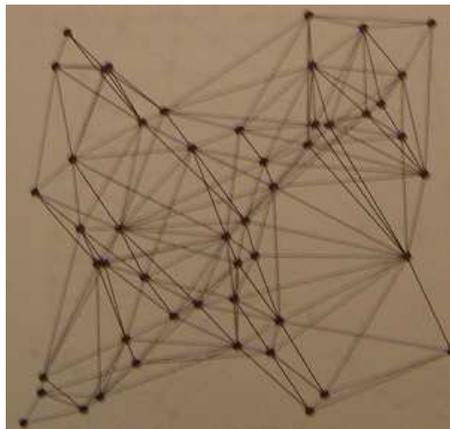
MST



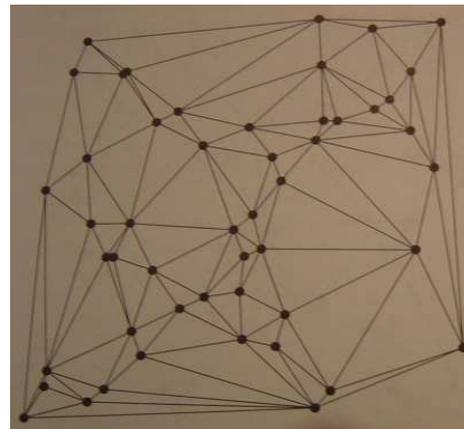
RNG



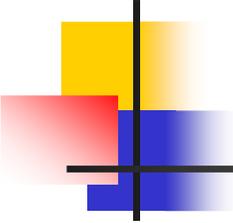
GG



YG

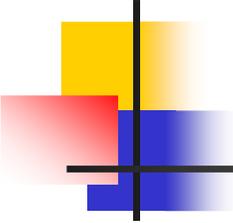


Del



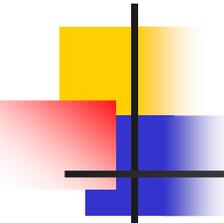
Estruturas Geométricas

- Em RSSF, é interessante projetar aplicações que utilizem algoritmos localizados:
 - Nós sensores apenas interagem com outros nós sensores numa vizinhança restrita
 - Coletivamente atingem um objetivo global
- $YG(V)$, $RNG(V)$, $GG(V)$
 - Podem ser construídos por algoritmos localizados
- $MST(V)$ e $Del(V)$
 - Não podem ser construídos por nenhum algoritmo localizado



Triangulação de Delaunay

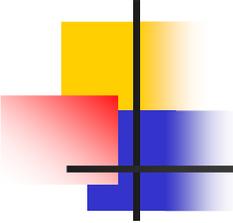
- T-Spanner
 - T aproximadamente 2,42
- Construção requer comunicação massiva
 - Não é apropriado para RSSF



Triangulação de Delaunay de Unidade

- $UDel(V)$
- É o grafo resultado de se remover todas as arestas de $Del(V)$ que são maiores do que uma unidade

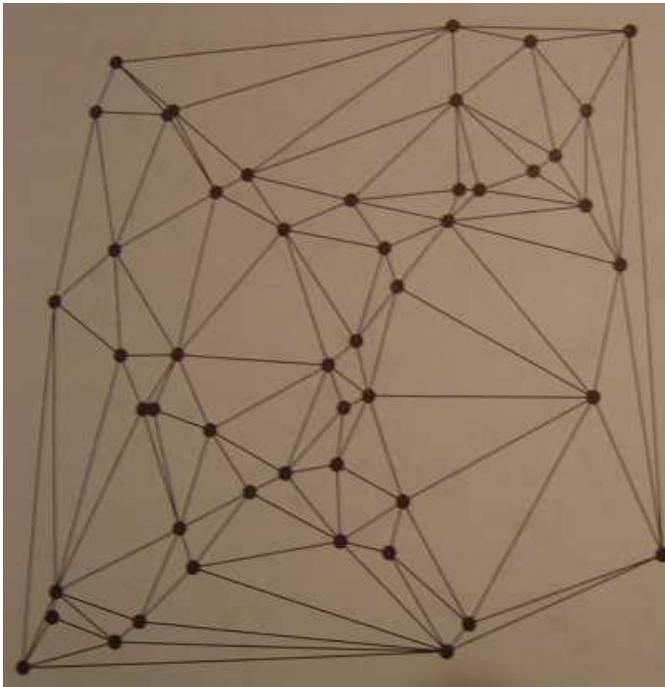
$$t = \frac{4\sqrt{3}}{9} \pi$$



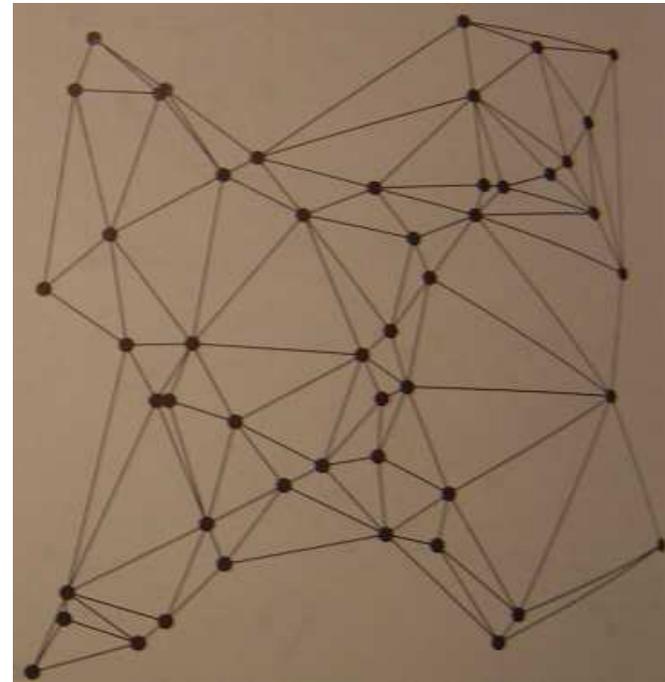
Triangulação de Delaunay Localizada

- $\text{LDel}^{(k)}(V)$
- Supergrafo de $\text{UDel}(V)$
- Construído por um algoritmo localizado
- Custo total de comunicação
 - $O(m \log n)$ bits
- $\text{PLDel}(V)$:
 - $O(n \log n)$ bits

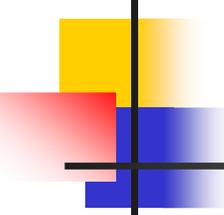
Triangulacao de Delaunay Localizada



Del(V)

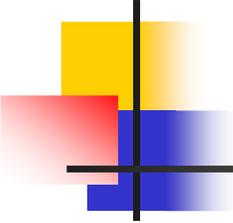


PLDel(V)



Triangulação de Delaunay Parcial

- PDT
- Contém um grafo de Gabriel como seu sub-grafo
- Sub-grafo de $UDel(V)$
- Construído por algoritmo localizado



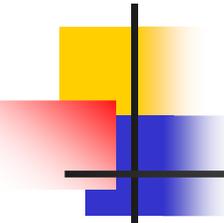
Geometria Computacional e RSSF

- Controle de Topologia
 - Objetivo: construir um sub-grafo (spanner) do UDG que seja esparso e possa ser construído localmente de maneira eficiente

Grafo esparso

X

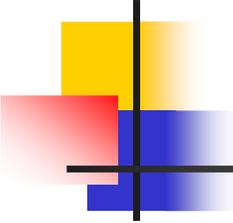
Gasto eficiente da energia



Geometria Computacional e RSSF

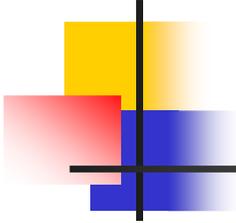
– Controle de Topologia

- Tolerância a falhas
 - Caminhos alternativos devem existir na topologia da rede
- Grafo deve ser esparso, mas isto não deve comprometer tolerância a falhas nem o consumo de energia



Geometria Computacional e RSSF

- Roteamento Localizado
 - Decisão sobre para onde enviar um pacote baseada em:
 - Informações no cabeçalho do pacote
 - Informações locais obtidas pelo nó em sua vizinhança



Geometria Computacional e RSSF

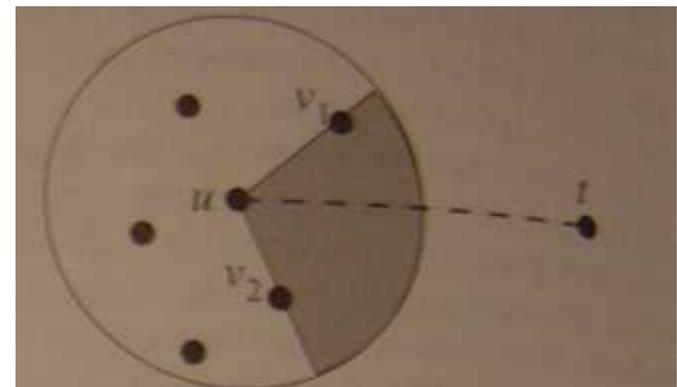
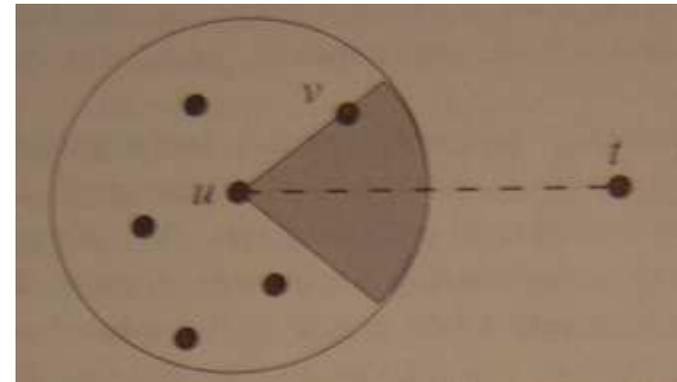
– Roteamento Localizado

- Nó de origem deve saber a localização aproximada do nó de destino
- Várias heurísticas
- Cenário:
 - Nó u quer transmitir para nó t

Geometria Computacional e RSSF

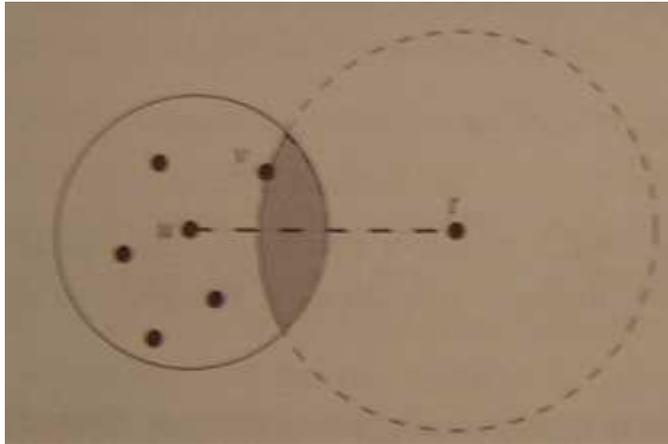
– Roteamento Localizado

- Compass routing:
 - u acha nó v de tal forma que o ângulo vut é o menor entre todos os vizinhos de u
- Random compass routing:
 - Similar ao compass routing
 - Diferença: dois pontos escolhidos, um acima e outro abaixo da linha ut . Escolhe aleatoriamente um dos dois

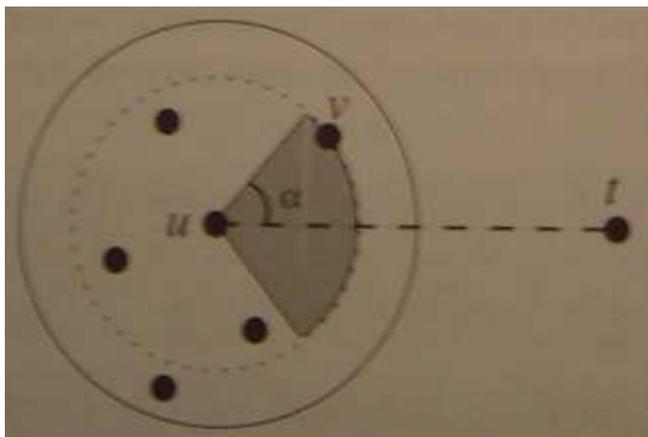


Geometria Computacional e RSSF

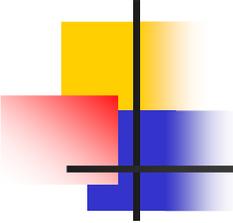
– Roteamento Localizado



- Greedy routing
 - u escolhe um nó v de tal forma que vt é o menor arco entre todos os vizinhos de u



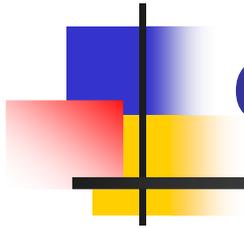
- Nearest neighbor routing
 - Dado um ângulo α , u escolhe o nó v mais próximo entre todos os vizinhos de u de tal forma que o ângulo $\angle vut$ seja menor ou igual a α

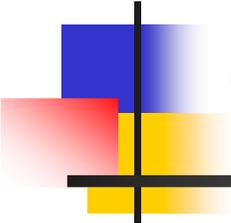


Geometria Computacional e RSSF

- Broadcasting
 - Objetivo: Minimizar o consumo de energia
 - Métodos centralizados
 - Não consideram overhead computacional e de comunicação
 - Assumem que topologia da rede não muda
 - Árvore geradora mínima (MST)
 - Árvore do caminho mínimo (Dijkstra)
 - Métodos localizados
 - Maioria baseada em MST distribuídas

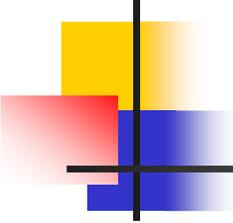
Aplicações – O que tem sido feito com Geometria Computacional em RSSF





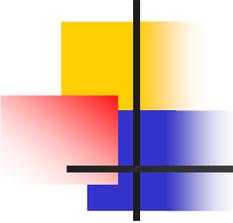
Problemas relacionados a Cobertura em RSSF

University of California, Los
Angeles 2001



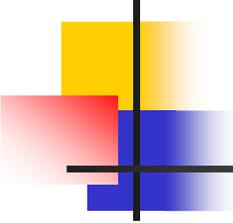
Objetivos da pesquisa

- Determinação da cobertura em caminhos da rede
 - Parâmetro de qualidade de serviço
 - Aumento da qualidade com colocação de novos nós ou reposicionamento dos existentes
 - Caminhos: Melhor e pior caso na cobertura



Algoritmo (1/2)

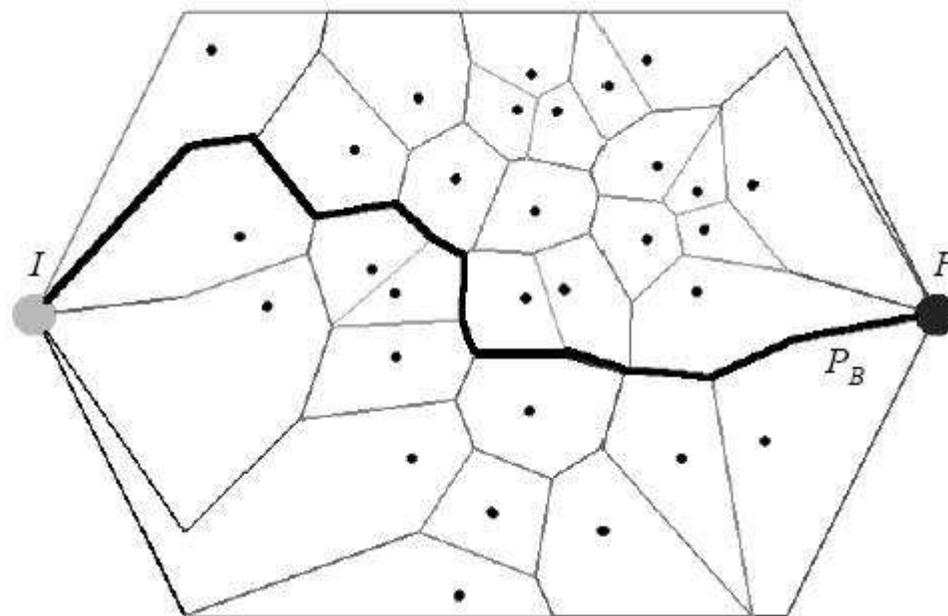
- Primeiro passo: algoritmo de localização geográfica
- Alguns nós sensores sabem a sua localização (beacons)
- Determinação da distância a esses nós por intensidade do sinal
- Por triangulação, nós descobrem a sua localização e se tornam novos beacons

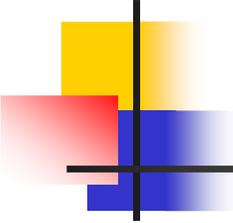


Algoritmo (2/2)

- Cálculo do diagrama de Voronoi e de triangulações de Delaunay
- Pior cenário: caminho de brecha máxima = segmentos do diagrama de Voronoi
- Melhor cenário: caminho de suporte máximo = segmentos dos triângulos de Delaunay

Pior cenário de cobertura

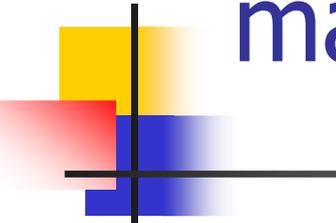




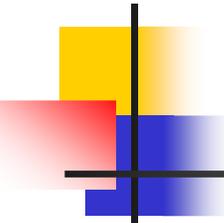
Procura do caminho de brecha máxima (1/2)

- Calcula Diagrama de Voronoi
- Transforma em grafo
- Procura segmentos que fazem parte do caminho (pior caso)
- Peso de cada aresta: distância até o sensor mais próximo

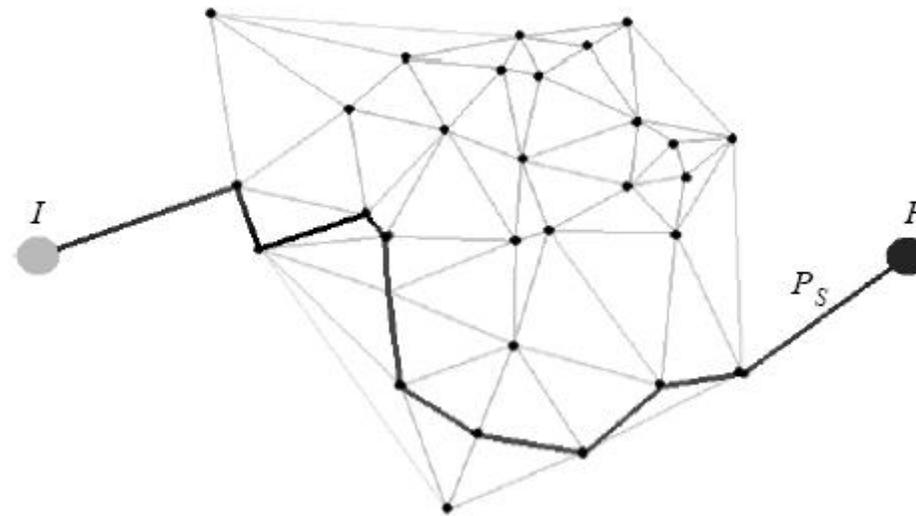
Procura do caminho de brecha máxima (2/2)

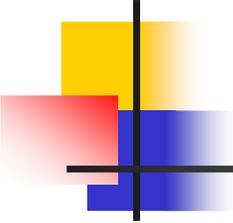


```
Generate Bounded Voronoi diagram for  $S$  with vertex set  $U$  and
line segment set  $L$ .
Initialize weighted undirected graph  $G(V,E)$ 
FOR each vertex  $u_i \in U$ 
    Create duplicate vertex  $v_i$  in  $V$ 
FOR each  $l_i(u_j, u_k) \in L$ 
    Create edge  $e_i(v_j, v_k)$  in  $E$ 
     $Weight(e_i) = \min$  distance from sensor  $s_i \in S$  for  $1 \leq i \leq |S|$ 
 $min\_weight = \min$  edge weight in  $G$ 
 $max\_weight = \max$  edge weight in  $G$ 
 $range = (max\_weight - min\_weight) / 2$ 
 $breach\_weight = min\_weight + range$ 
WHILE ( $range > binary\_search\_tolerance$ )
    Initialize graph  $G'(V', E')$ 
    FOR each  $v_i \in V$ 
        Create vertex  $v_i'$  in  $G'$ 
    FOR each  $e_i \in E$ 
        IF  $Weight(e_i) \geq breach\_weight$ 
            Insert edge  $e_i'$  in  $G'$ 
         $range = range / 2$ 
    IF  $BFS(G', I, F)$  is Successful
         $breach\_weight = breach\_weight + range$ 
    ELSE
         $breach\_weight = breach\_weight - range$ 
    END IF
```



Melhor cenário de cobertura

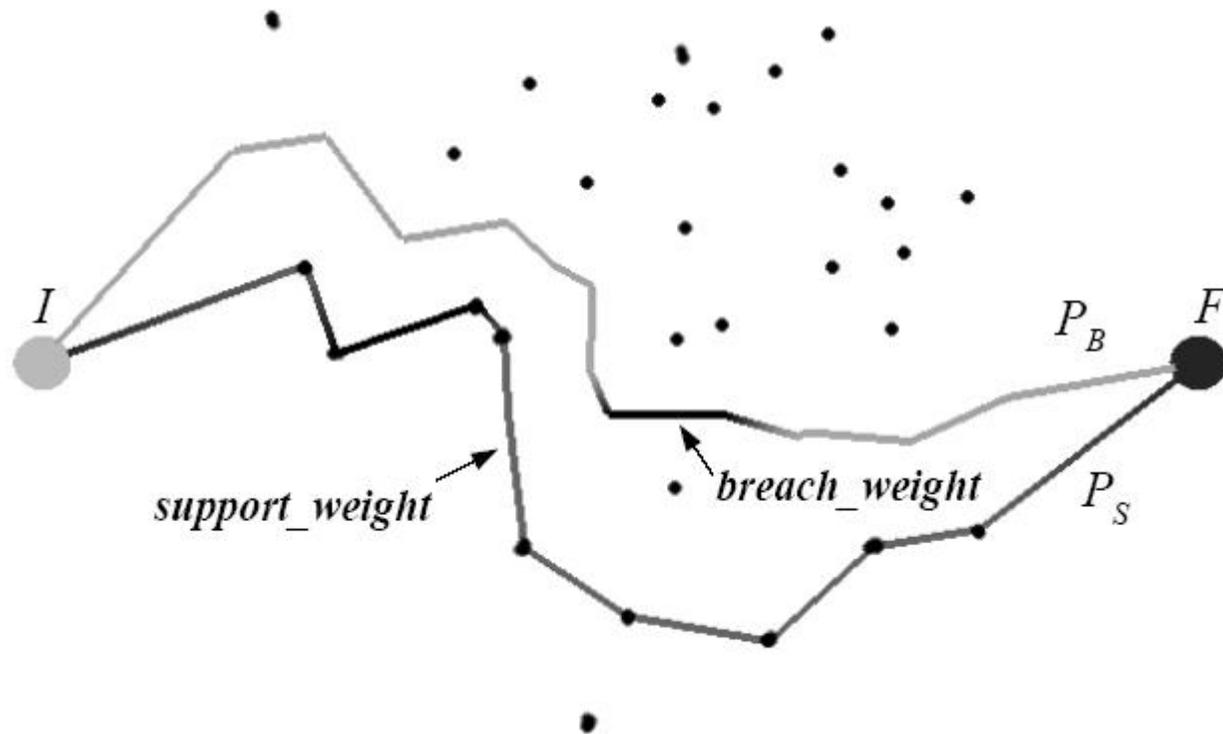




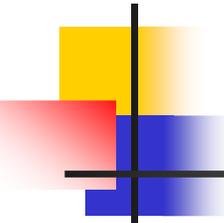
Procura do caminho de suporte máximo

- Algoritmo semelhante, porém:
 - troca-se o diagrama de Voronoi pelas triangulações de Delaunay
 - Peso da aresta é o tamanho do segmento
 - *breach_weight* é substituído por *support_weight*

Pior cenário e melhor cenário

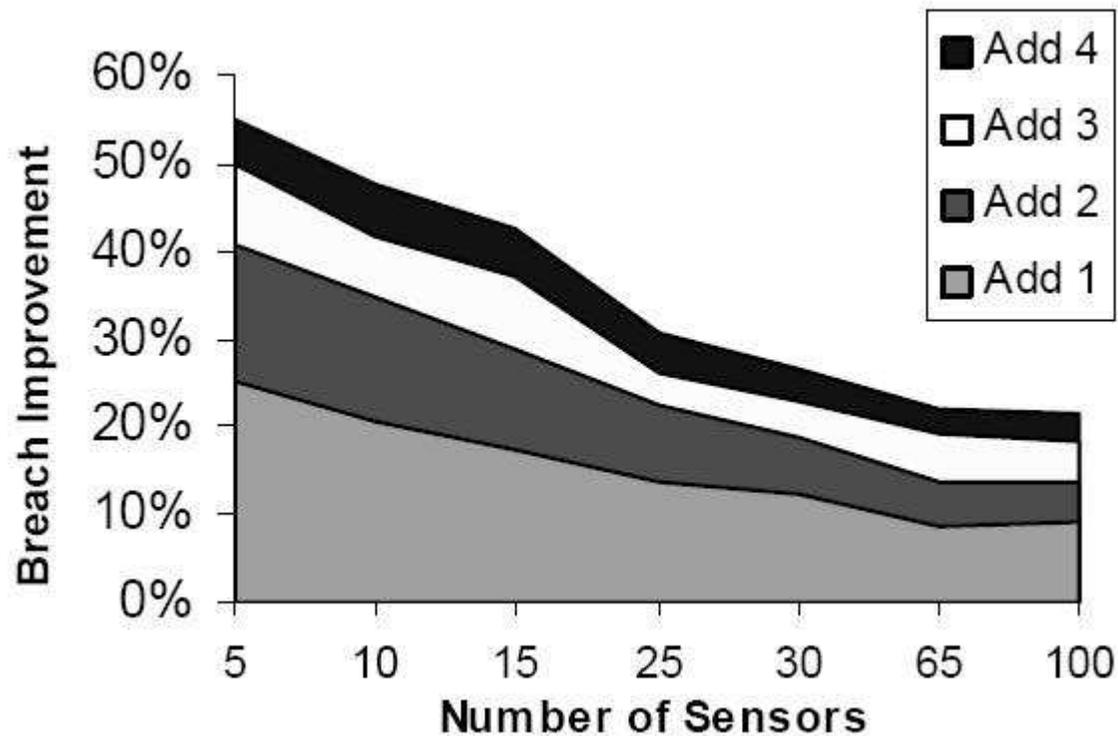


Heurísticas de adição de sensores

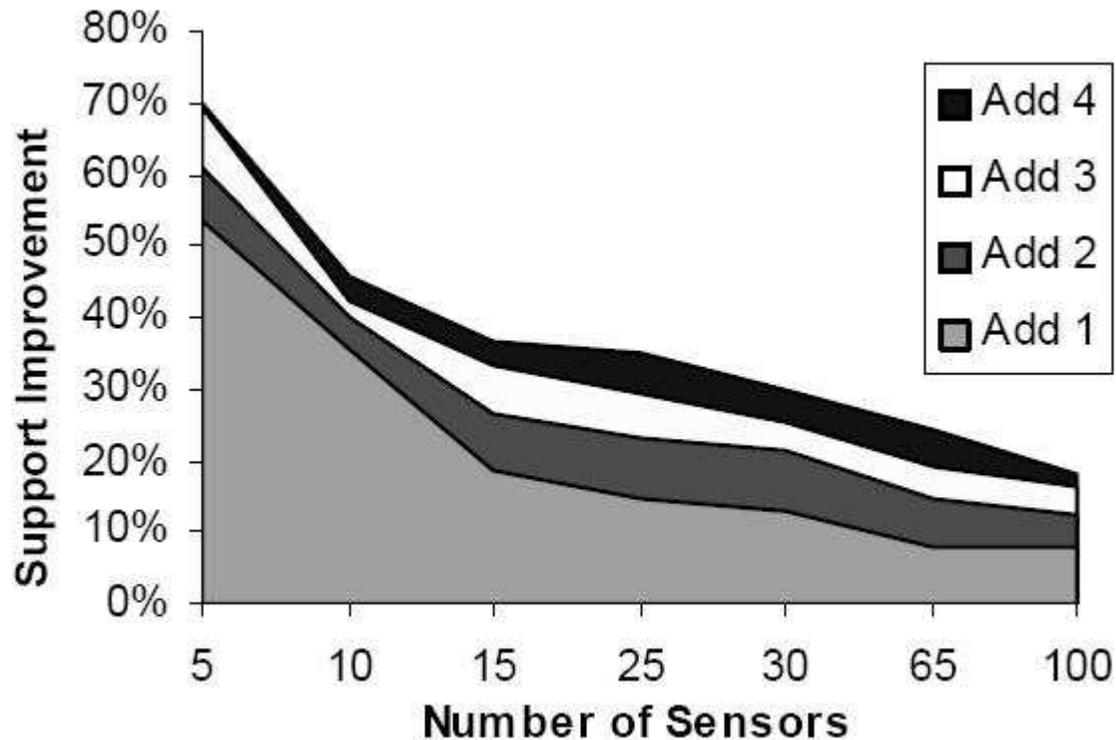


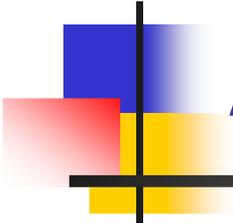
- Adição de sensores ao longo da aresta que tem menor peso (pior cenário)
- Adição de sensores no ponto médio da aresta com maior peso (melhor cenário)

Ganho com adiç o de sensores no caminho de pior caso



Ganho com adiç3o de sensores no caminho de melhor caso



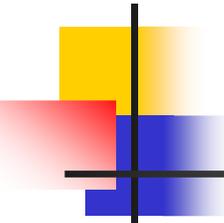


Agregação em RSSF

University of California, Los
Angeles 2004

Aggregação Nodal x Espacial

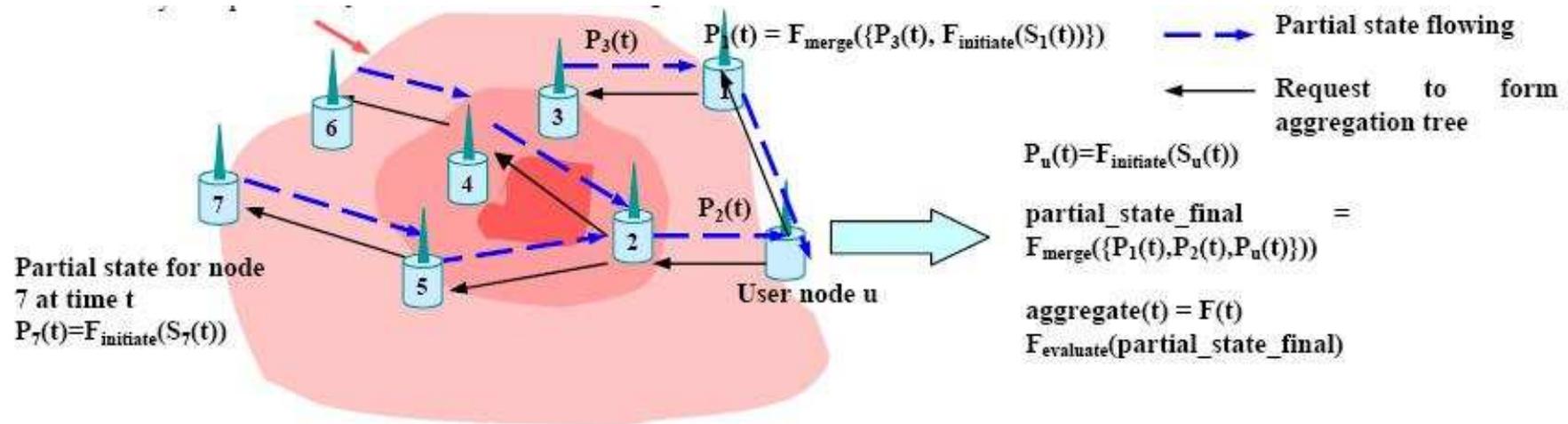
(1/3)



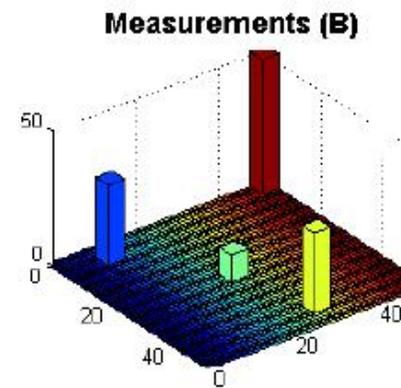
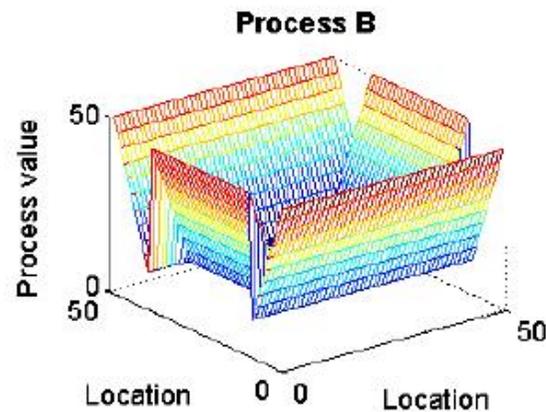
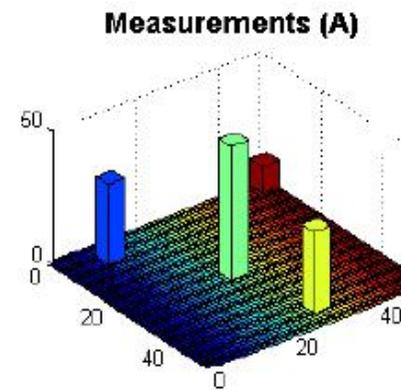
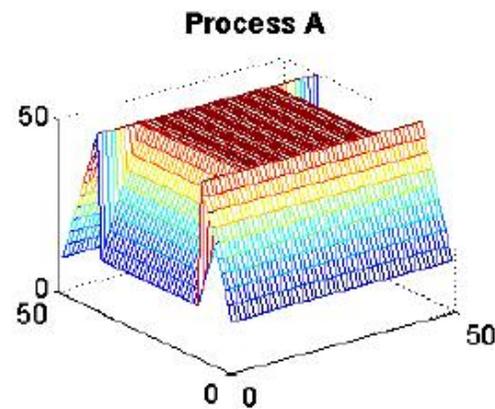
- Usuário não está interessado em dados dos nós, mas de **uma região**.
- Ao invés de “me dê a temperatura média nos nós X, Y e Z”:
- **“Me dê a temperatura média nessa sala”**

Agregação Nodal x Espacial (2/3)

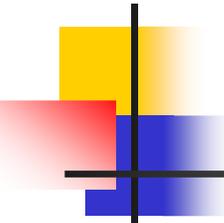
- Funções: *Initiate*, *merge*, *evaluate*



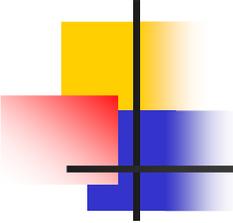
Aggregação Nodal x Espacial (3/3)



Cálculo do diagrama de Voronoi

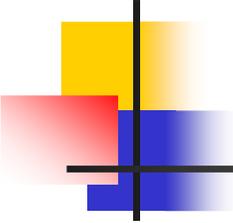


- Centralizado
 - Exato, os dados de todos os nós são processados em um servidor central
- Distribuído
 - Aproximado, pois não há conhecimento de TODOS os nós



Centralizado

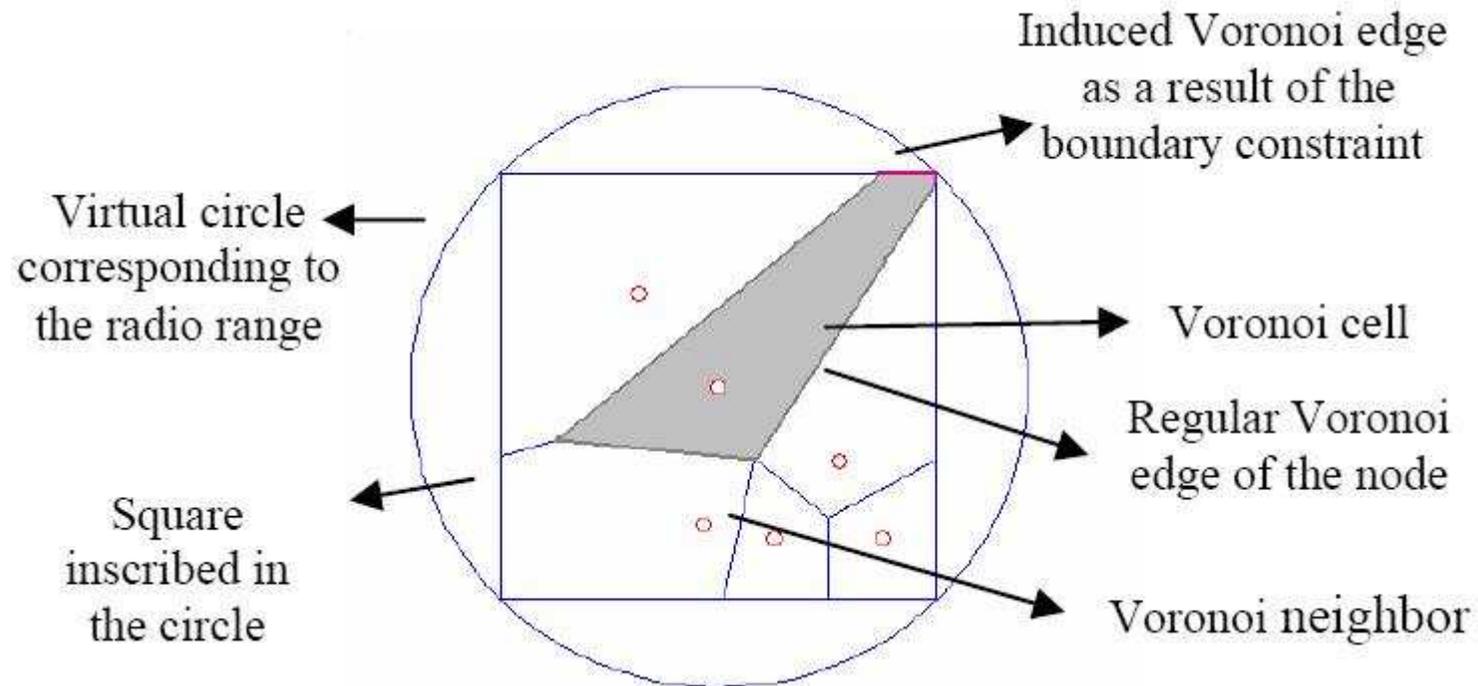
1. Cada sensor envia sua localização para o servidor. O servidor calcula então a célula de Voronoi para cada nó.
2. O servidor envia de volta aos nós a área da célula de Voronoi de cada um deles.
3. A agregação é feita com o peso equivalente à área da célula de Voronoi



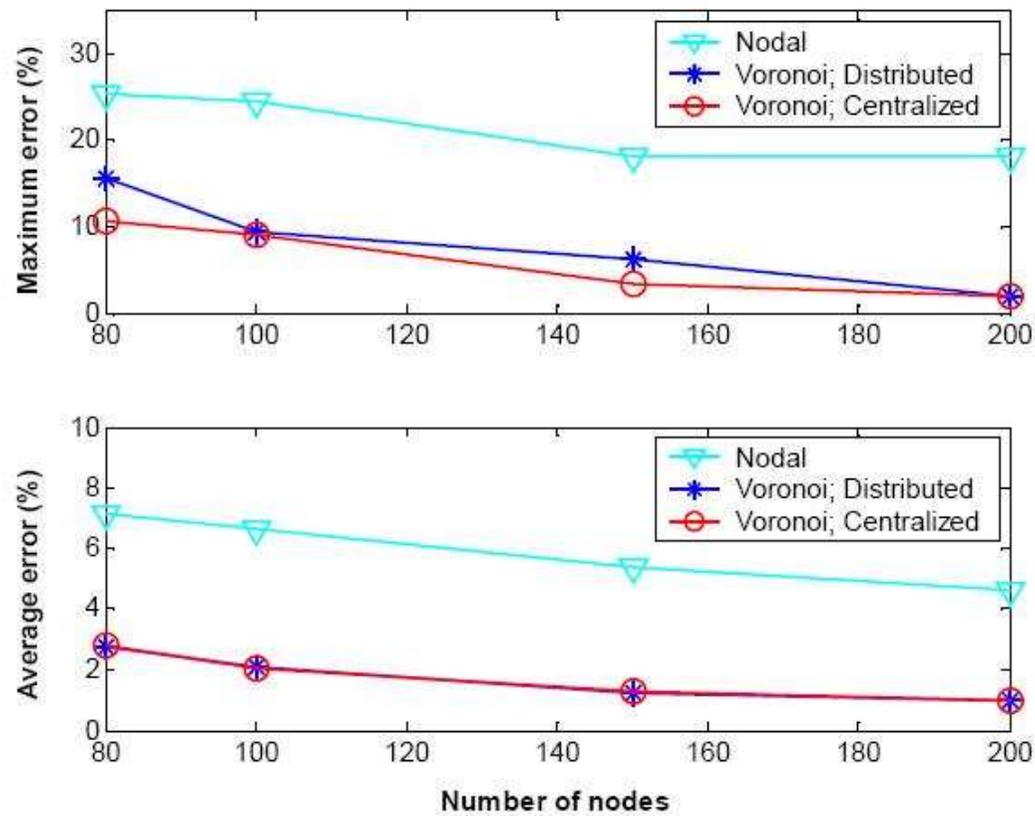
Distribuído (1/2)

- Apenas vizinhos imediatos são levados em conta no cálculo
- Mas, vizinhos no diagrama podem estar a mais de um hop de distância!
- Limite da área: quadrado inscrito no círculo determinado pelo alcance do rádio do nó

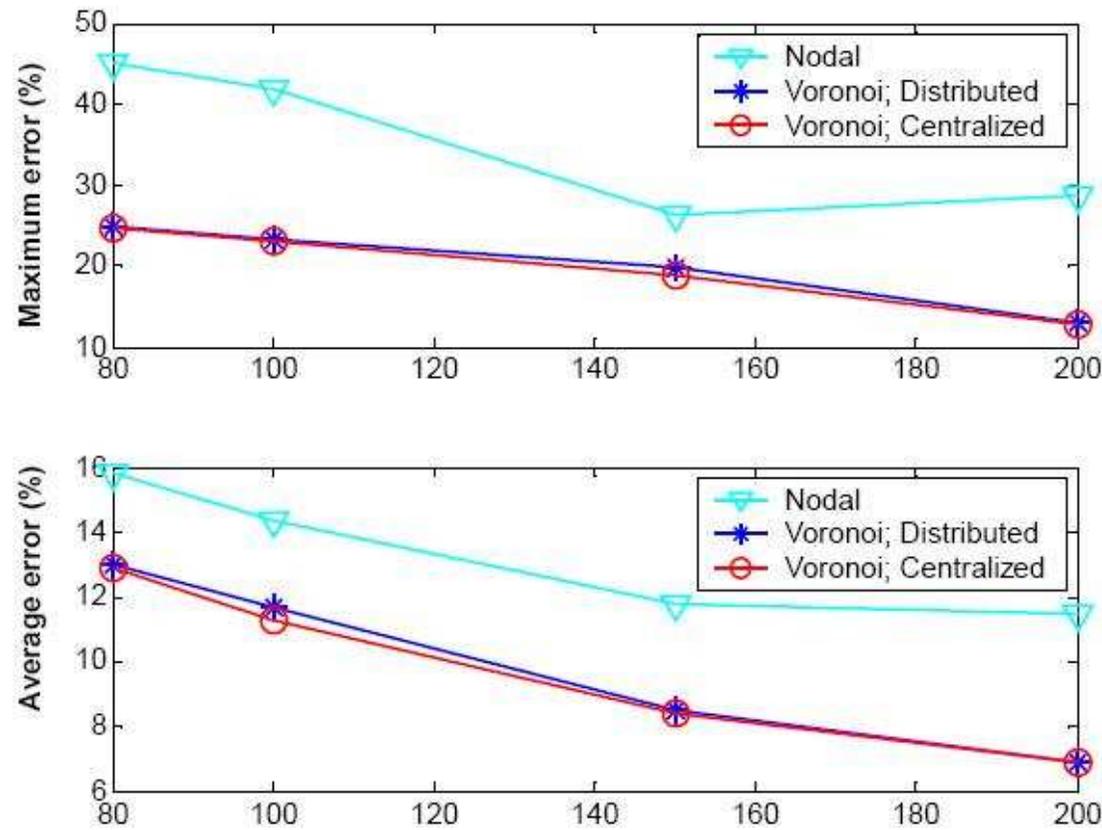
Distribuído (2/2)

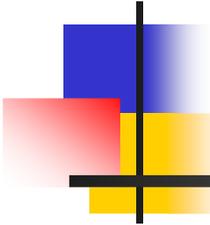


Resultados – Cálculo da Média

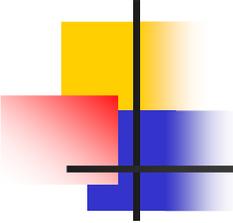


Resultados – Cálculo de histograma



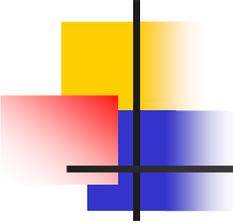


Conclusões



Conclusões

- Geometria computacional aplica-se diretamente a RSSF
- Apresenta soluções
- Apresenta também novos desafios:
 - Processamento dos algoritmos
 - Distribuição dos algoritmos



Referências

- **Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems**

Edited by: Mohammad Ilyas and Imad Mahgoub

- **Coverage Problems ins Wireless Ad-hoc Sensor Networks, INFOCOM 2001**

http://www.cs.ucla.edu/~miodrag/papers/Meguerdichian_Infocom_01.pdf

- **Going beyond nodal aggregation in Sensor Networks, NESL Technical Report, August 2004**

<http://www.ee.ucla.edu/~saurabh/publications/TECH-REPORT-VIAS.pdf>