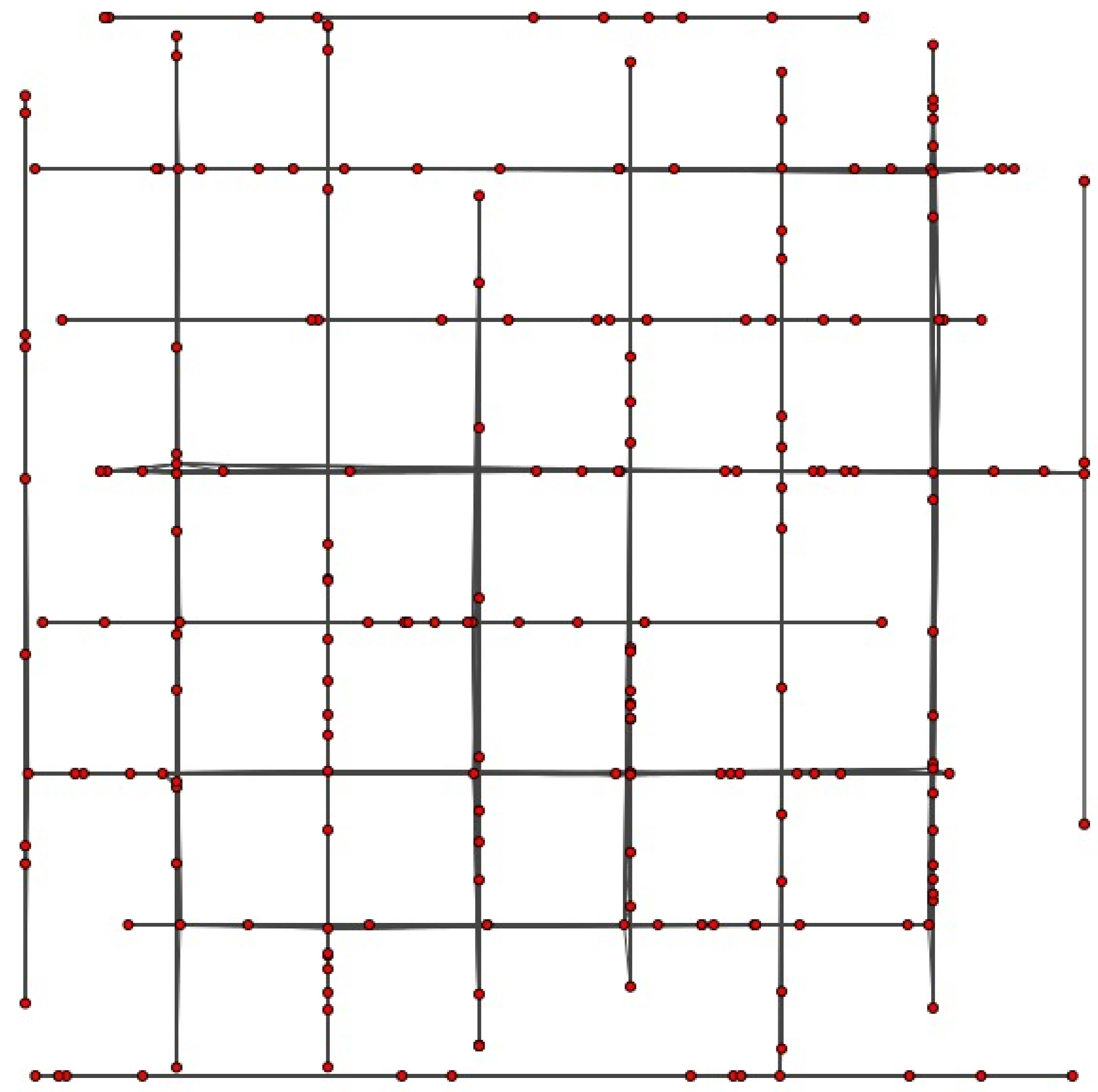
Bottleneck Steiner Tree (BST)

Manasses Ferreira

manassesferreira@dcc.ufmg.br

Federal University of Minas Gerais (UFMG)

Wireless Informational Sensing Embedded systems Models Algorithms and Protocols (WISEMAP)



Descrição do Problema Prático

Bottleneck Steiner Tree $\mathcal{BST}$ Dado um conjunto $\mathcal{T} = \{t_1, t_2, ..., t_n\}$ de n **Terminais** e um inteiro positivo k, obtenha a **Árvore de Steiner** contendo, no máximo, k **Pontos de Steiner** $\mathcal{S} = \{s_1, s_2, ..., s_k\}$ tal que o comprimento da aresta mais longa seja minimizado.[3]

Terminais Vértices localizados em uma dada posição no plano.

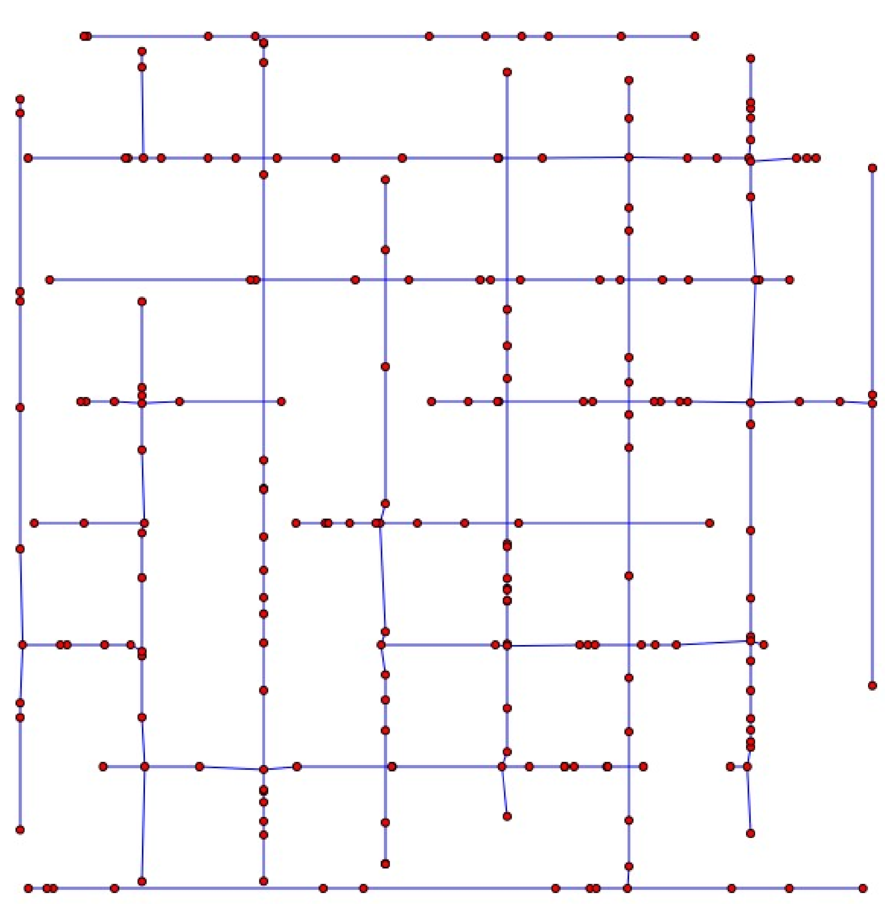
Árvore de Steiner Rede de comunicação acíclica entre os Terminais.

Pontos de Steiner Vértices _não-terminais_ da Árvore de Steiner.

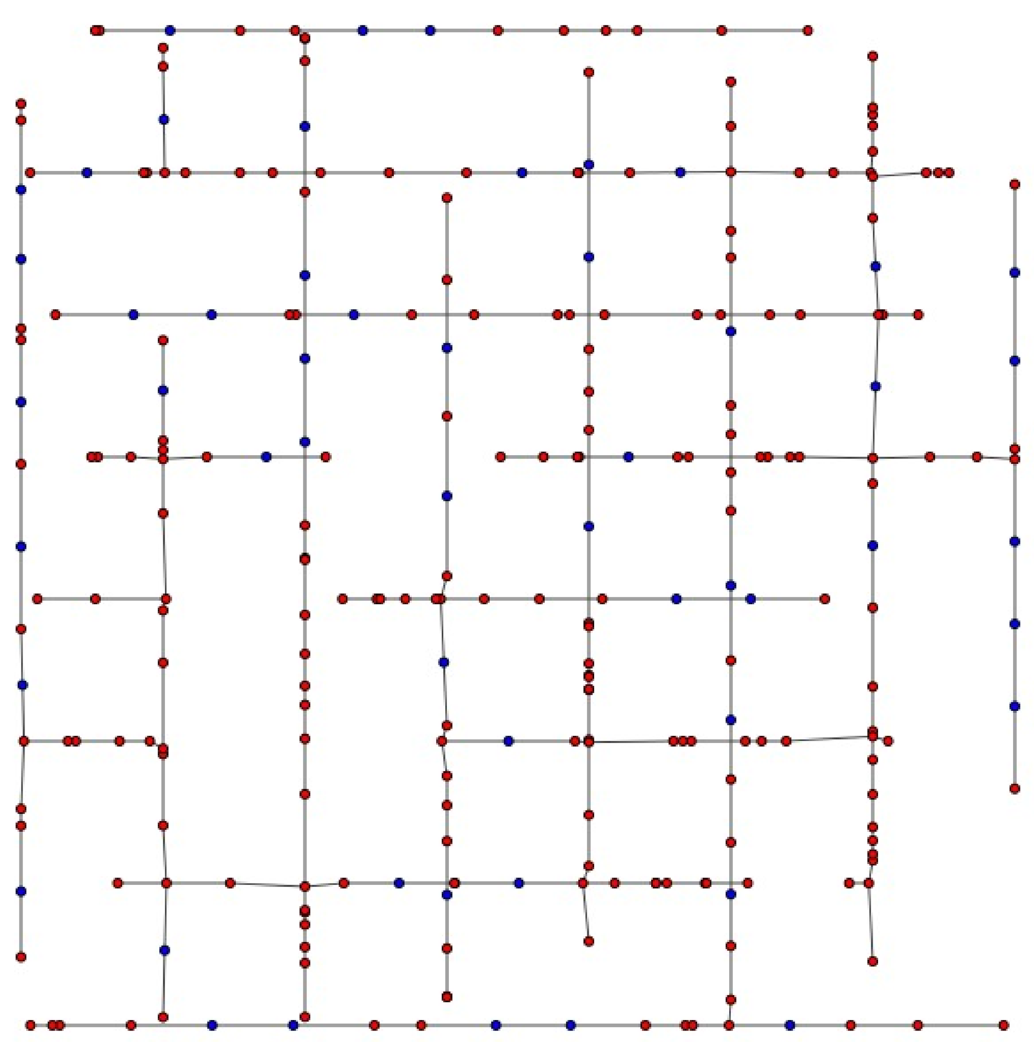
Algorithm 2: SteinerizedSpanningTree

Input: Grafo planar euclidiano $G(\mathcal{T}, E)$ e um inteiro k
Output: Árvore geradora steinerizada T

- $T(\mathcal{S}, \mathcal{E}) \leftarrow$ MinimumSpanningTree(G) $\mid$ $\mathcal{S} \leftarrow \mathcal{T}$, $\mathcal{E}_{\mathcal{S}} \subseteq E$, $n \leftarrow |\mathcal{T}|$
- **for** $e \leftarrow 1$ **to** $|\mathcal{E}|$ **do**
  - // $(v_i, v_j) \leftarrow e$ $\mid$ $v_i, v_j \in \mathcal{S}$, $e \in \mathcal{E}_{\mathcal{S}}$, $i \neq j$, $i, j \in \mathbb{N}_{>0}$
  - $\mathcal{P}_e \leftarrow \varnothing$
  - $p_e \leftarrow 0$
  - $l_e \leftarrow w_e$ // $w_e \rightarrow [0, \infty)$
- **for** $s \leftarrow n+1$ **to** $n+k$ **do**
  - $e \leftarrow \arg\max_{i \in [1, n-1]} \; l_i$
  - $\mathcal{P}_e \leftarrow \mathcal{P}_e \cup \{s\}$
  - $p_e \leftarrow p_e + 1$
  - $l_e \leftarrow \frac{w_e}{p_e + 1}$
- **for** $e \leftarrow 1$ **to** $|\mathcal{E}|$ **do**
  - **if** $p_e > 0$ **then**
    - $(v_i, v_j) \leftarrow e$ // $i, j \in [1, n]$
    - $\alpha \leftarrow$ FIRST$(\mathcal{P}_e)$
    - $\mathcal{E} \leftarrow \mathcal{E} - \{e\} \cup \{(v_i, v_\alpha)\}$
    - **for** _each_ $s \in \mathcal{P}_e$ **do**
      - $\mathcal{S} \leftarrow \mathcal{S} \cup \{v_s\}$ // $s \in [n+1, n+k]$
      - **if** $\nu \leftarrow$ NEXT$(s, \mathcal{P}_e)$ **then**
        - $\mathcal{E}_{\mathcal{S}} \leftarrow \mathcal{E}_{\mathcal{S}} \cup \{(v_s, v_\nu)\}$
    - $\omega \leftarrow$ LAST$(\mathcal{P}_e)$
    - $\mathcal{E}_{\mathcal{S}} \leftarrow \mathcal{E}_{\mathcal{S}} \cup \{(v_\omega, v_j)\}$
- **return** T



MST



MST_SST

Algorithm 1: SteinerMinimalTree

Input: Grafo planar euclidiano $G(\mathcal{T}, E)$
Output: Árvore de Steiner Minimal T^* e um inteiro k

1. $\{T^*, k\} \leftarrow$ GeoSteiner(G)
2. **return** $\{T^*, k\}$

	Total Length	Bottleneck
MST	90.36	2.35
MST_SST	90.36	0.68
SMT	61.87	0.95
SMT_SST	61.87	0.38

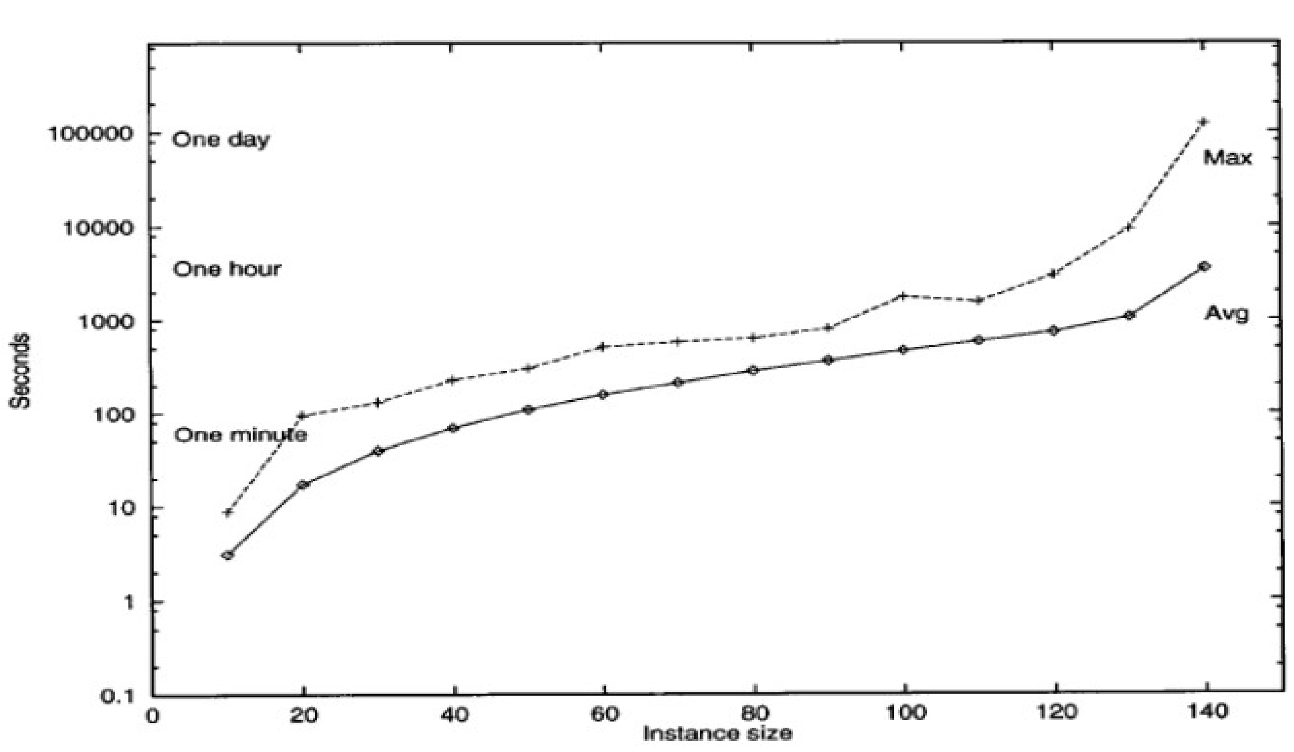
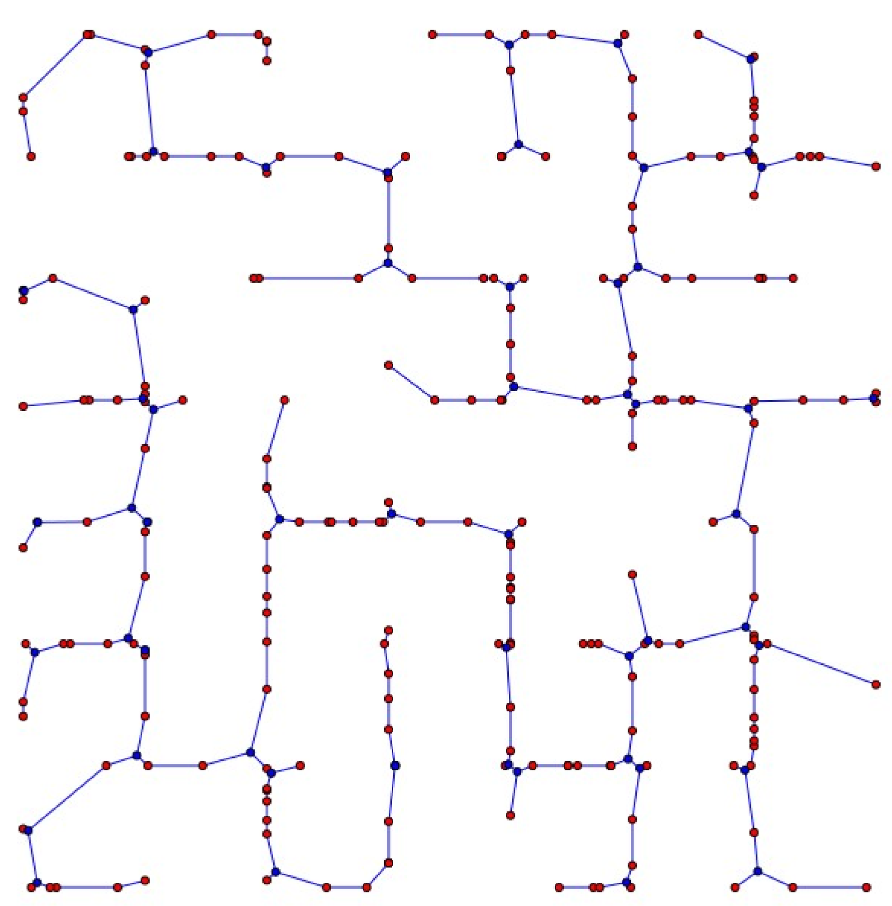
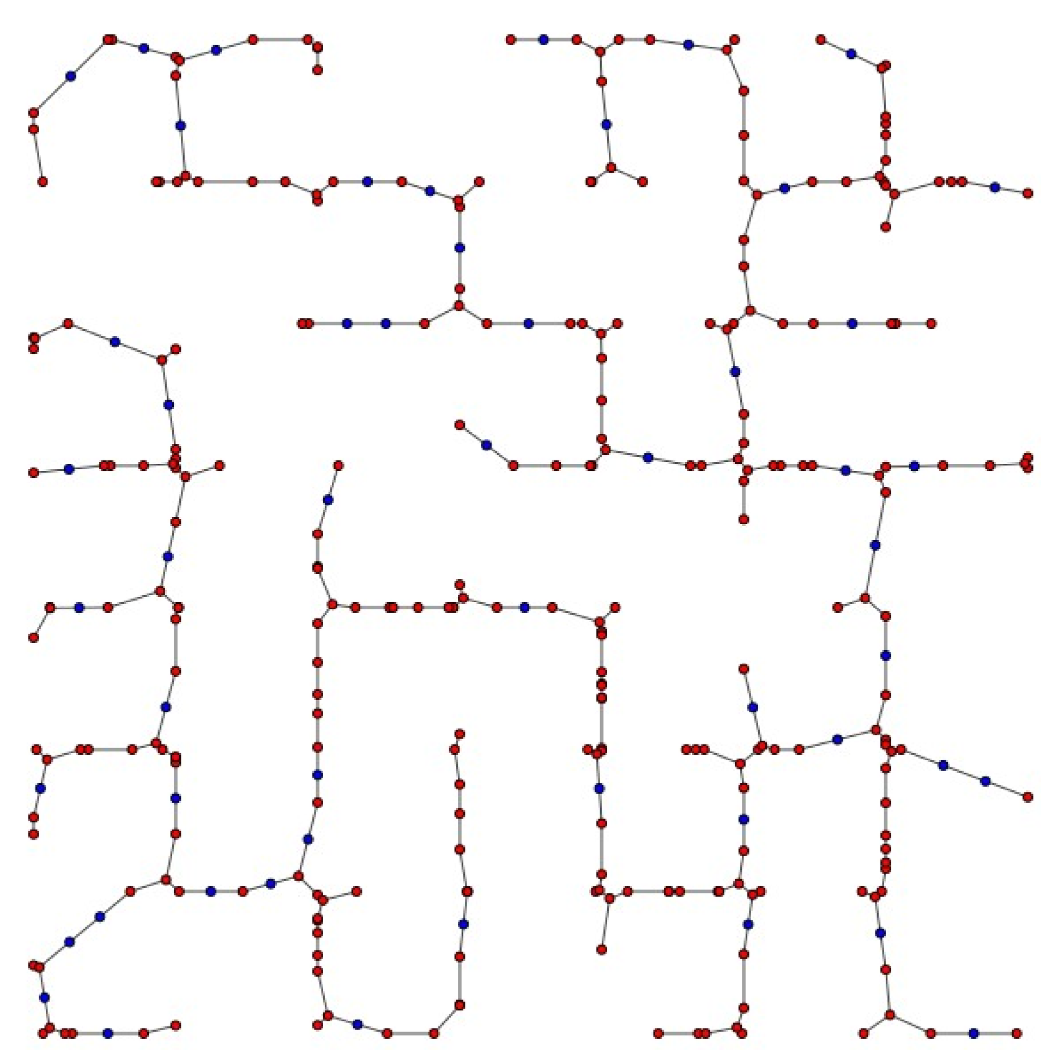


Fig. 11. Total CPU times (seconds).



SMT



SMT_SST

WISEMAP