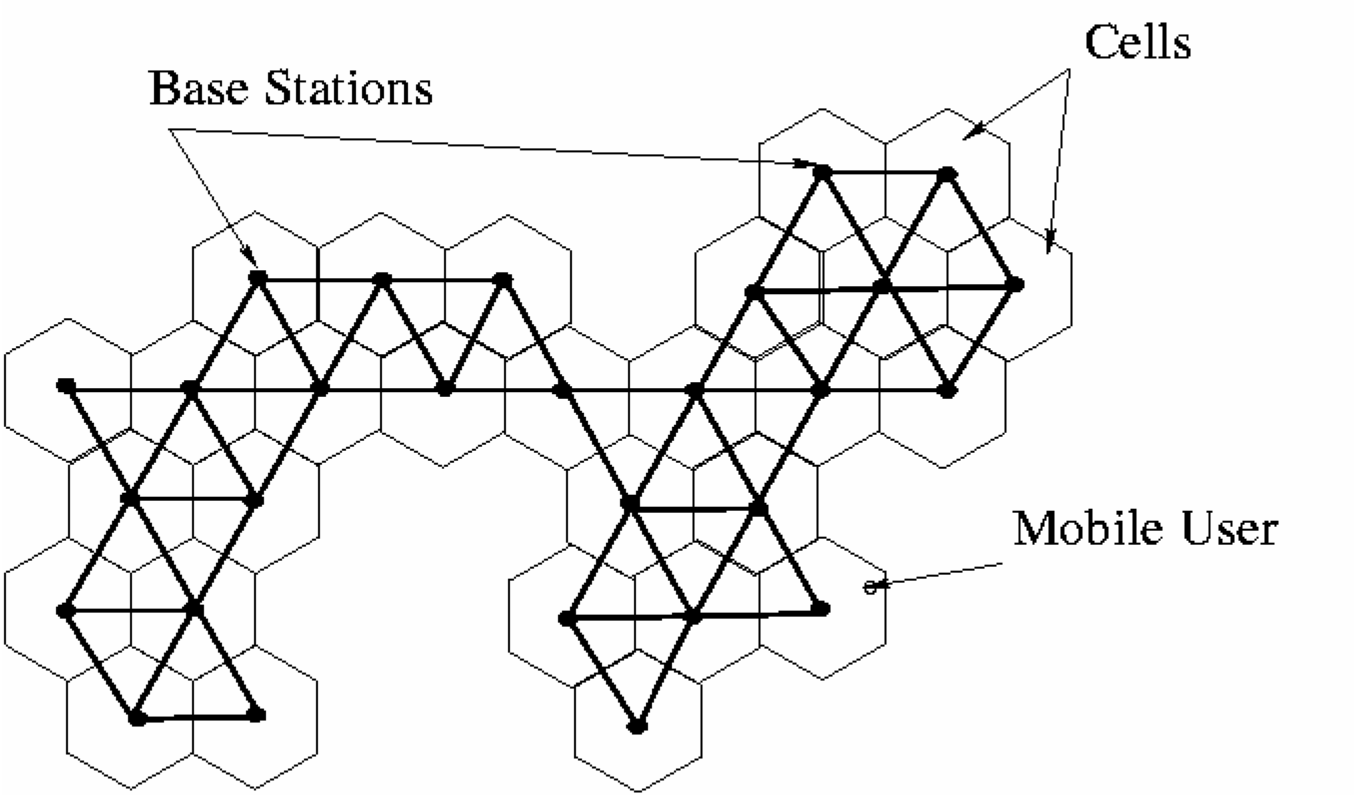
Distance-Based Location Update and Routing in Irregular Cellular Networks

Victor Chepoi, Feodor Dragan, Yan Vaxes

University of Marseille, France
Kent State University, Ohio, USA

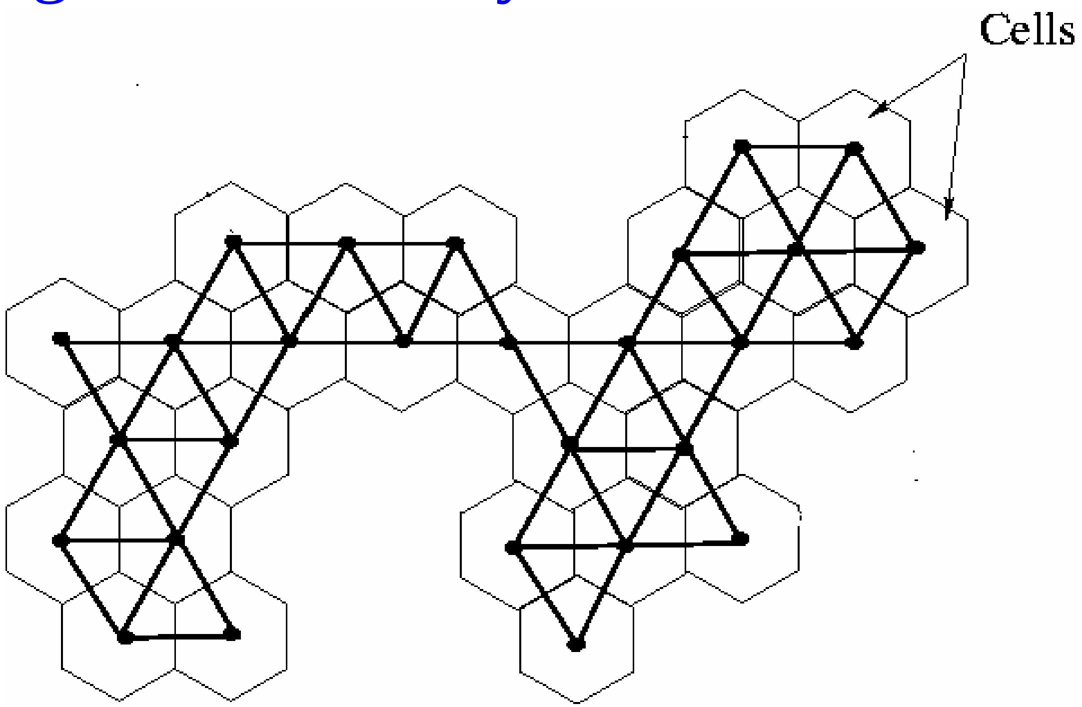
SAWN 2005

Regular Cellular Network



Regular Cellular Network as Benzenoid and Triangular Systems

- Benzenoid Systems: is a simple circuit of the hexagonal grid and the region bounded by this circuit.



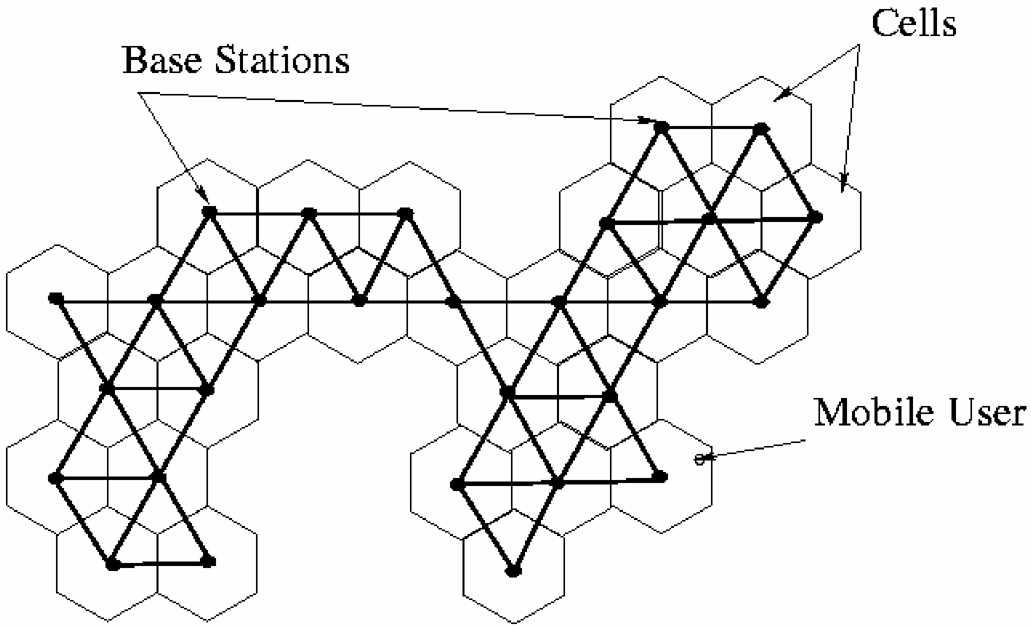
- The Duals to Benzenoid Systems are Triangular Systems

Addressing, Distances and Routing: Necessity

- Identification code (CIC) for tracking mobile users
- Dynamic location update (or registration) scheme
  - time based
  - movement based
  - distance based

(cell-distance based is best, according to [Bar-Noy&Kessler&Sidi'94])

- → Distances
- Routing protocol

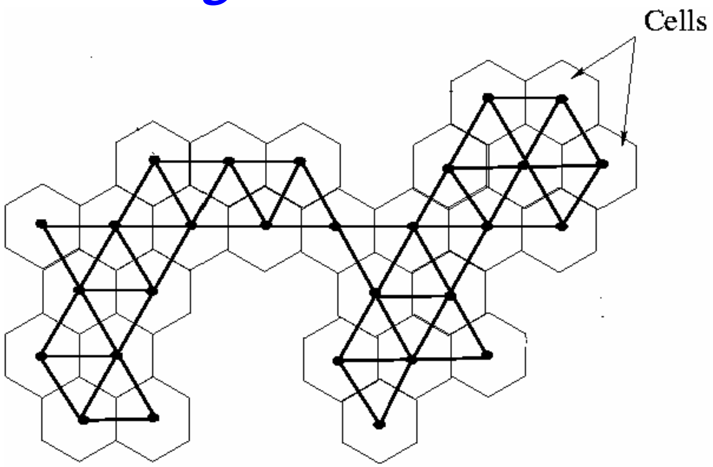


Current situation

- Current cellular networks do not provide information that can be used to derive cell distances
  - It is hard to compute the distances between cells (claim from [Bar-Noy&Kessler&Sidi'94])
  - It requires a lot of storage to maintain the distance information among cells (claim from [Akyildiz&Ho&Lin'96] and [Li&Kameda&Li'00])

Our WMAN'04 results for triangular systems

- Scale 2 isometric embedding into Cartesian product of 3 trees
  - ➔ cell addressing scheme using only three small integers
  - ➔ distance labeling scheme with labels of size $O(\log^2 n)$ -*bits per node and constant time distance decoder*
  - ➔ routing labeling scheme with labels of size $O(\log n)$*-bits per node and constant time routing decision.*



Distance Labeling Scheme

Goal: Short labels that encode distances and distance decoder, an algorithm for inferring the distance between two nodes only from their labels (in time polynomial in the label length)

- **Labeling:** *v* → *Label(v)*

  (for trees, $O(\log^2 n)$ *bits per node* [Peleg'99]*)*

- **Distance decoder:** *D(Label(v), Label(u))* → *dist(u,v)*

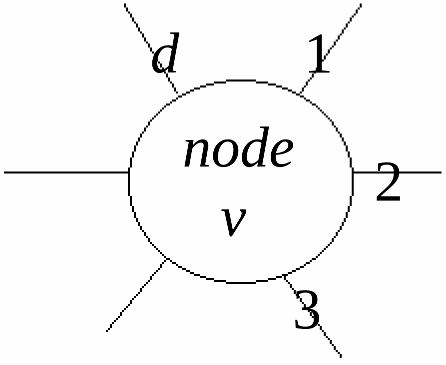
  (for trees, constant decision time)

Routing Labeling Scheme

Goal: Short labels that encode the routing information and routing protocol, an algorithm for inferring port number of the first edge on a shortest path from source to destination, giving only labels and nothing else

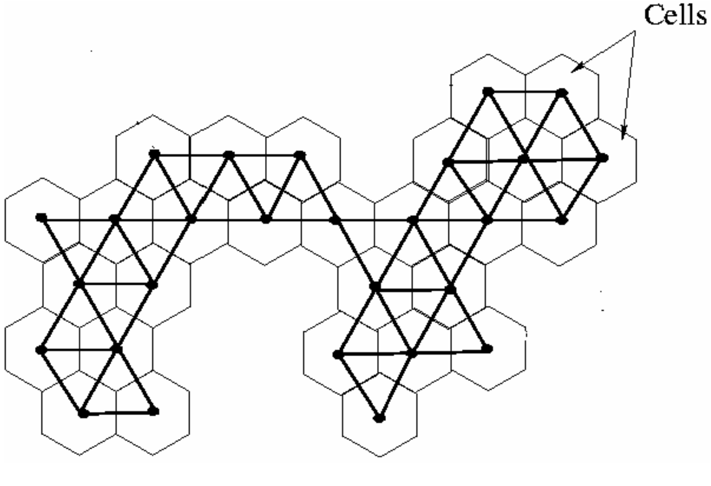
- **Labeling:** *v* → *Label(v)*
- **Distance decoder:** *R(Label(v), Label(u))* → *port(v,u)*

(for trees, $O(\log n)$ *bits per node and constant time decision* [Thorup&Zwick'01]*)*

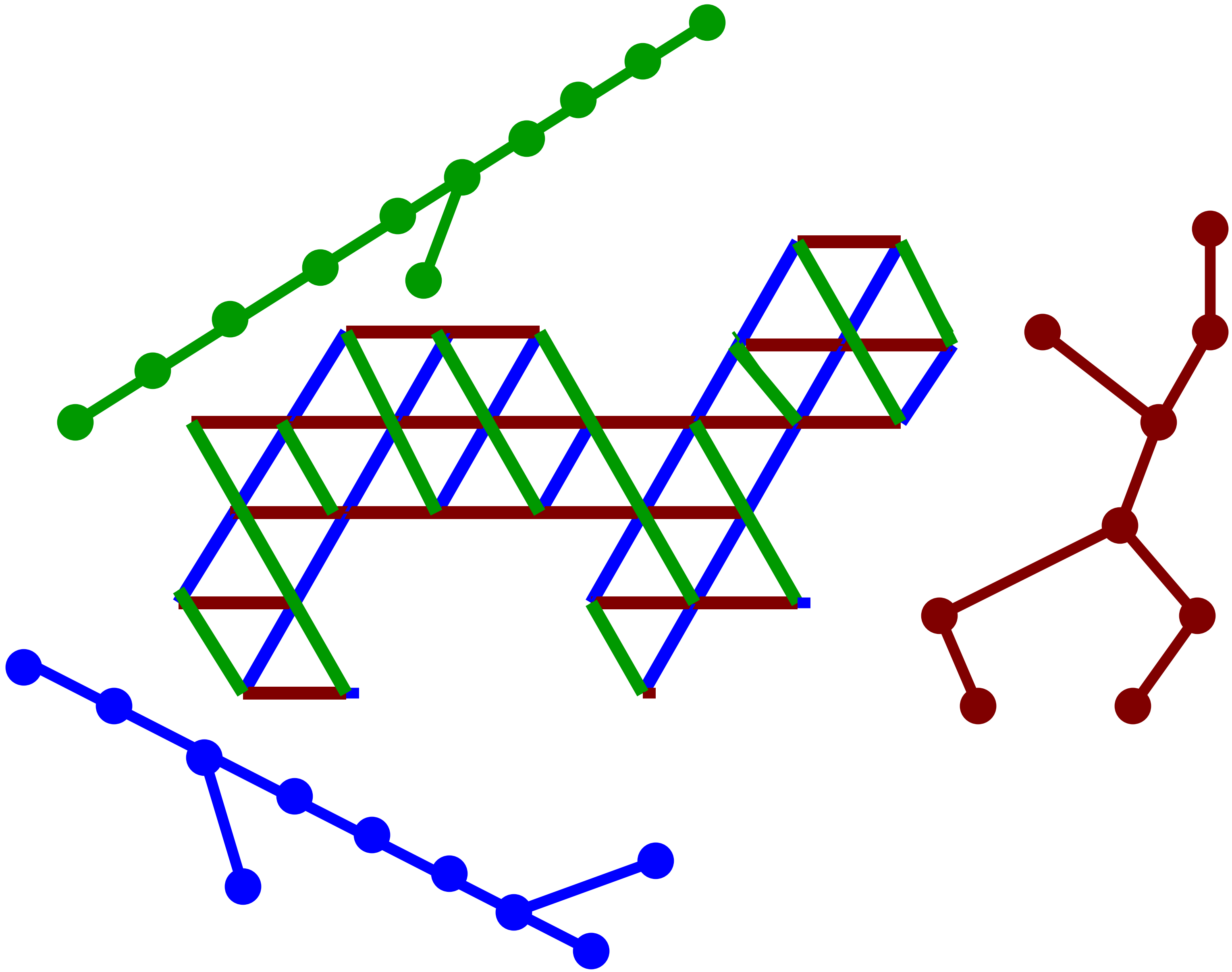


Our WMAN'04 results for triangular systems

- Scale 2 isometric embedding into Cartesian product of 3 trees
  - ➔ cell addressing scheme using only three small integers
  - ➔ distance labeling scheme with labels of size $O(\log^2 n)$ -*bits per node and constant time distance decoder*
  - ➔ routing labeling scheme with labels of size $O(\log n)$*-bits per node and constant time routing decision.*

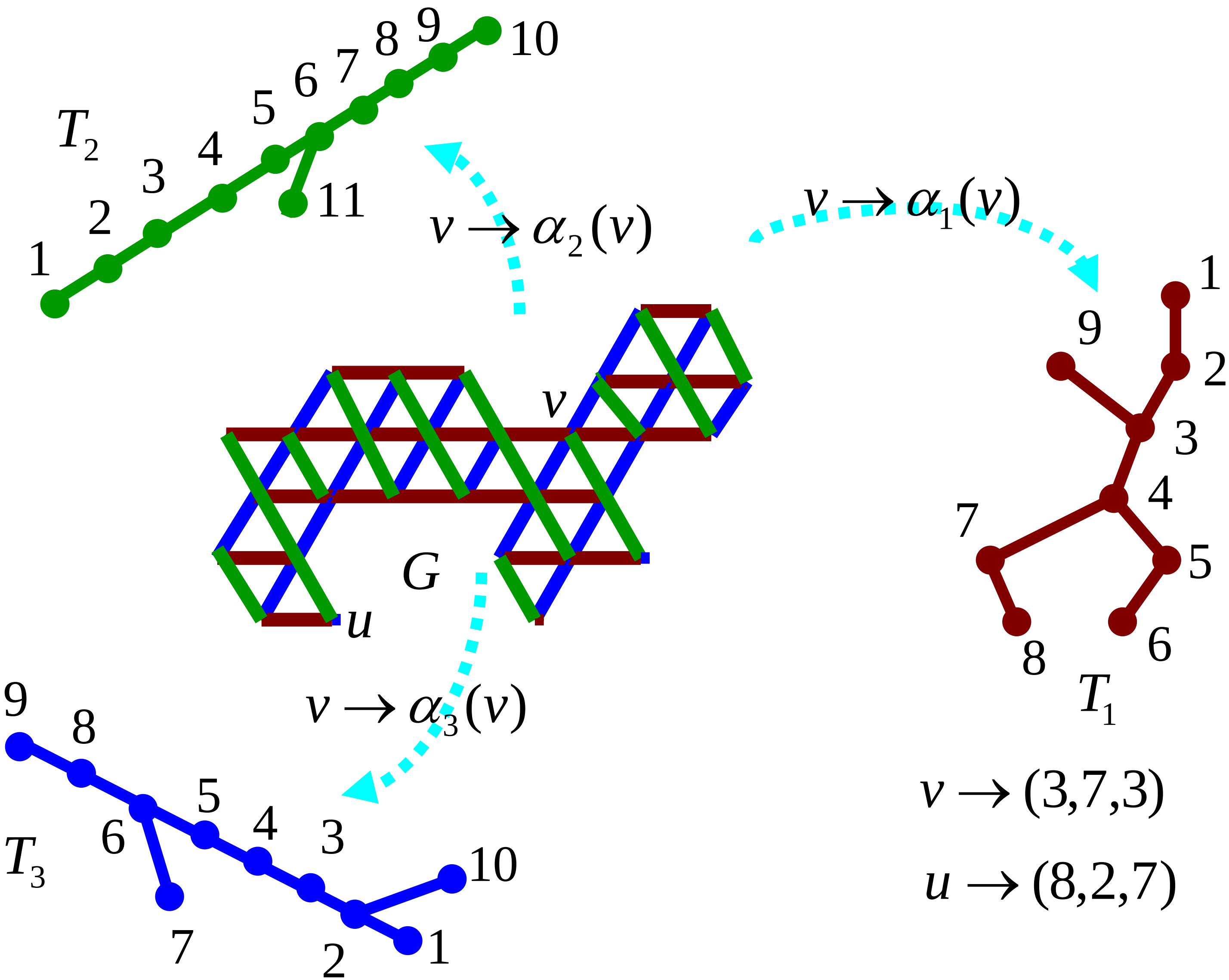


Three edge directions → three trees



Addressing

$$v \to (\alpha_1(v), \alpha_2(v), \alpha_3(v))$$

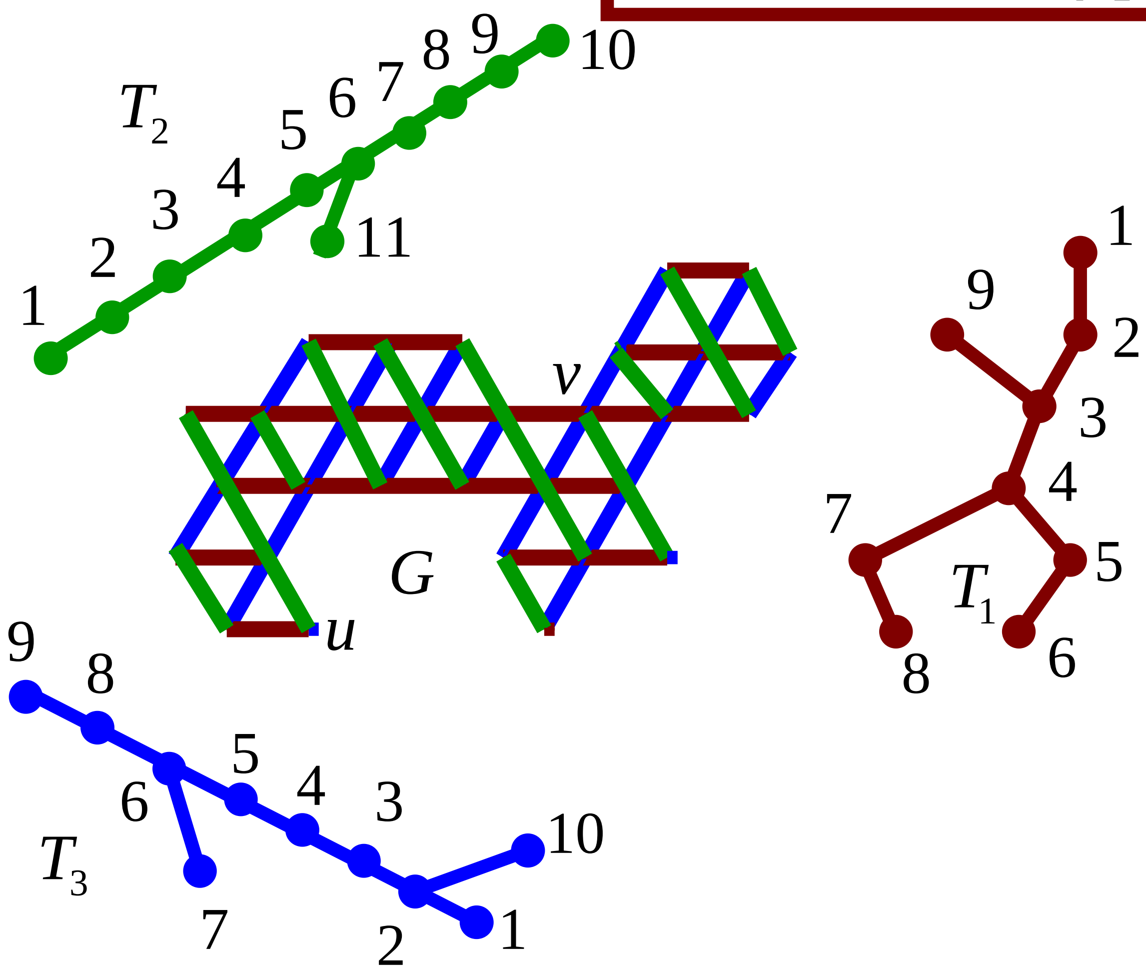


Scale 2 embedding into 3 trees

$$v \to (\alpha_1(v), \alpha_2(v), \alpha_3(v))$$

$$u \to (\alpha_1(u), \alpha_2(u), \alpha_3(u))$$

$$2dist_G(v,u) = \sum_{i=1}^{3} dist_{T_i}(\alpha_i(v), \alpha_i(u))$$



$$v \to (3,7,3)$$

$$u \to (8,2,7)$$

$$dist_{T_1}(\alpha_1(v), \alpha_1(u)) +$$
$$dist_{T_2}(\alpha_2(v), \alpha_2(u)) +$$
$$dist_{T_3}(\alpha_3(v), \alpha_3(u)) =$$
$$3 + 5 + 4 = 12 =$$
$$2dist_G(v,u)$$

Distance labeling scheme for triangular systems

- Given *G*, find three corresponding trees T_1, T_2, T_3 and addressing $v \to (\alpha_1(v), \alpha_2(v), \alpha_3(v))$ (*O(n) time*)

- Construct distance labeling scheme for each tree
  $\alpha_i(v) \to Label(\alpha_i(v))$ (*O(n*log*n) time*)

- Then, set $Label(v) = (Label(\alpha_1(v)), Label(\alpha_2(v)), Label(\alpha_3(v)))$

- $O(\log^2 n)$ -bit labels and constructible in total time $O(n \log n)$

Distance decoder for triangular systems

Given *Label(u)* and *Label(v)*

Function distance_decoder_triang_syst(Label(u),Label(v))

- Output ***½(distance_decoder_trees(*** $Label(\alpha_1(v)), Label(\alpha_1(u))$ ***)***

  +(distance_decoder_trees($Label(\alpha_2(v)), Label(\alpha_2(u))$ ***)***

  +(distance_decoder_trees($Label(\alpha_3(v)), Label(\alpha_3(u))$ ***))***

Thm: The family of *n*-node triangular systems enjoys a distance labeling scheme with $O(\log^2 n)$-bit labels and a constant time distance decoder.

Routing labeling scheme for triangular systems

- Given *G*, find three corresponding trees T_1, T_2, T_3 and addressing $v \rightarrow (\alpha_1(v), \alpha_2(v), \alpha_3(v))$

- Construct routing labeling scheme for each tree using Thorup&Zwick method (*log n* bit labels)

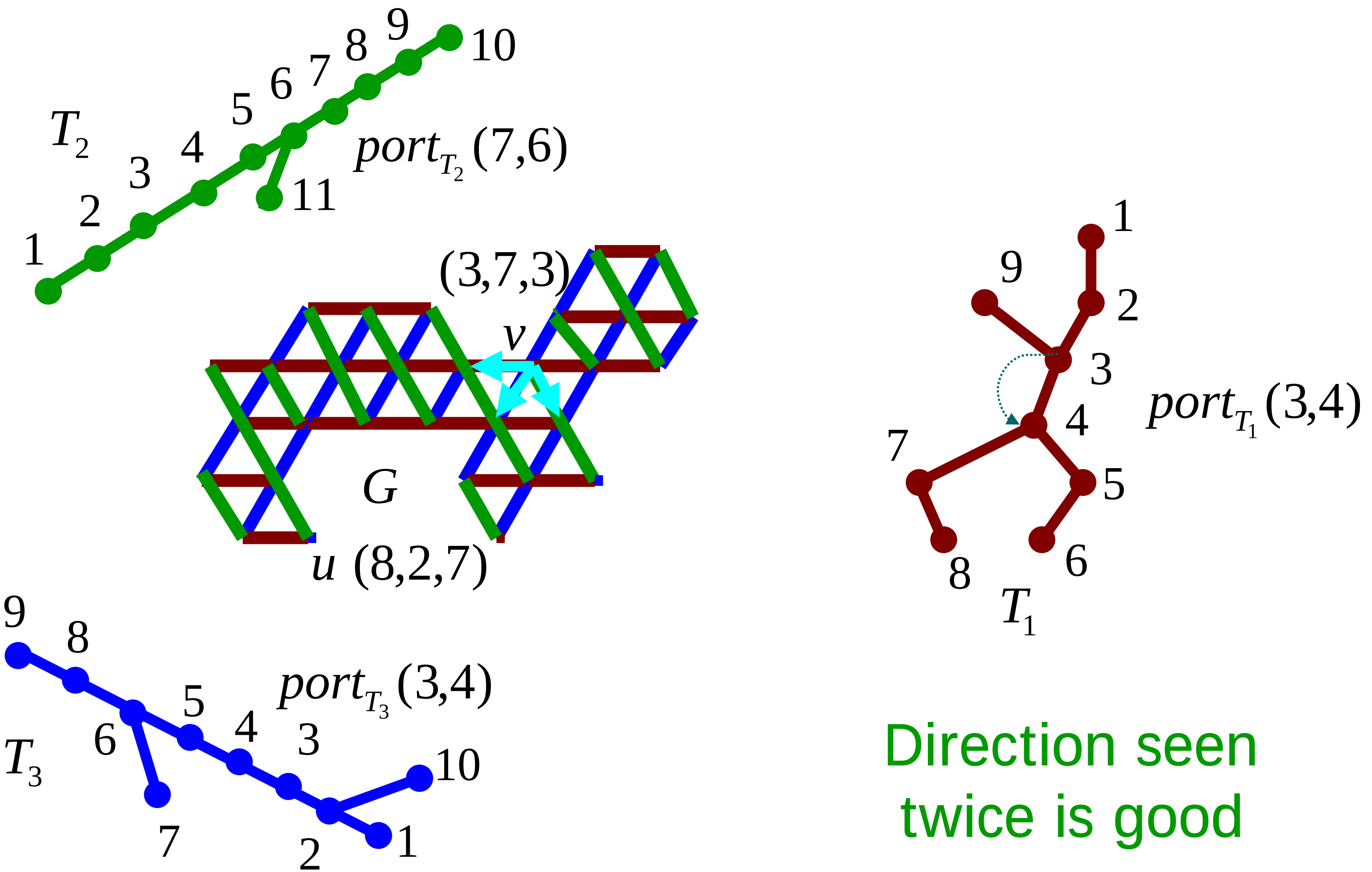
$$\alpha_i(v) \rightarrow Label(\alpha_i(v))$$

- Then, set

$$Label(v) = (Label(\alpha_1(v)), Label(\alpha_2(v)), Label(\alpha_3(v)), \ldots.)$$

Something more

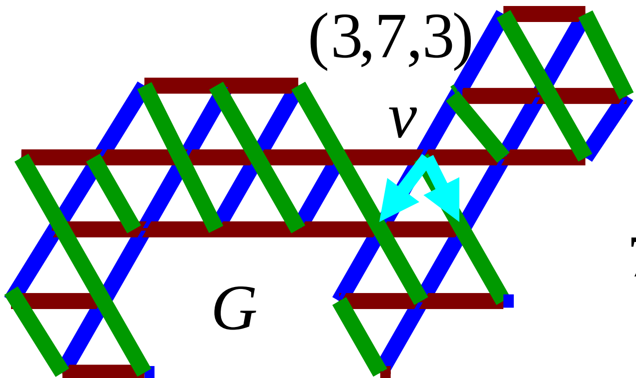
Choosing direction to go from v

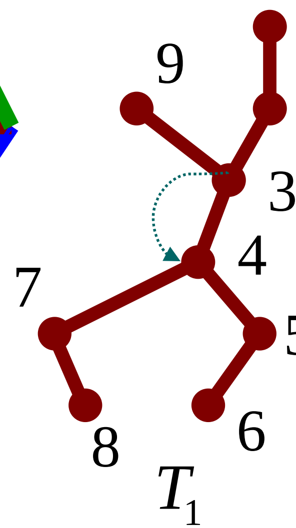


Direction seen twice is good

Mapping tree ports to graph ports

$$\Delta_i(v) = (\alpha_i^1(v),\ldots,\alpha_i^j(v)) = (2,4) \quad (j \le 4)$$





$port_{T_1}(3,4)$

$$O_i(v) = ((port_{T_i}(\alpha_i(v),\alpha_i^1(v)), Q_i^1(v)),\ldots,(port_{T_i}(\alpha_i(v),\alpha_i^j(v)), Q_i^j(v)) =$$

$$((port_{T_1}(3,2), Q_1^1(v)),\ldots,(port_{T_1}(3,4), Q_1^2(v))$$

$$Q_i^j(v) = (port_G^1, port_G^2)$$

Then, $Label(v) = (Label(\alpha_1(v)), Label(\alpha_2(v)), Label(\alpha_3(v)),\ldots.)$

(i.e., *3xlog n+3x4x3xlogn* bit labels)

Routing Decision for triangular systems

Given *Label(u)* and *Label(v)*

```
function routing_decision_triang_syst(L(x), H(y))

if (α1(x), α2(x), α3(x)) = (α1(y), α2(y), α3(y)) then return "packet reached its destination";
set A ← 0;
for each i ∈ {1,2,3} do
   p ← routing_decision_trees(L_Ti(αi(x)), H_Ti(αi(y)));
   for each j ∈ {1,...,|Oi(x)|} do
      if p = Oi(x)[j] then
         for each entry port_G of the array Q_i^j(x) do
            A[port_G] ← A[port_G] + 1;
            if A[port_G] = 2 then
               return port_G.
```

Thm: The family of *n*-node triangular systems enjoys a routing labeling scheme with $O(\log n)$-bit labels and a constant time routing decision.

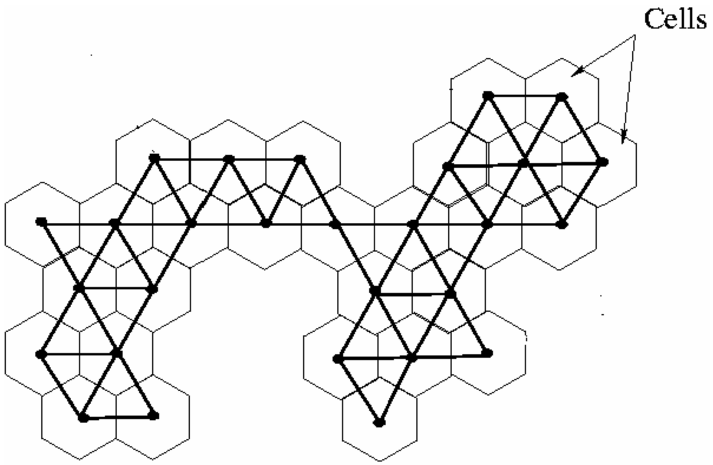
Cellular Networks in Reality

- Planned as uniform configuration of BSs, but in reality BS placement may not be uniformly distributed (← obstacles)

- To accommodate more subscribers, cells of previously deployed cellular network need to be split or rearranged into smaller ones.

- The cell size in one area may be different from the cell size in another area (dense/sparse populated areas)

- Very little is known for about cellular networks with non-uniform distribution of BSs and non-uniform cell sizes

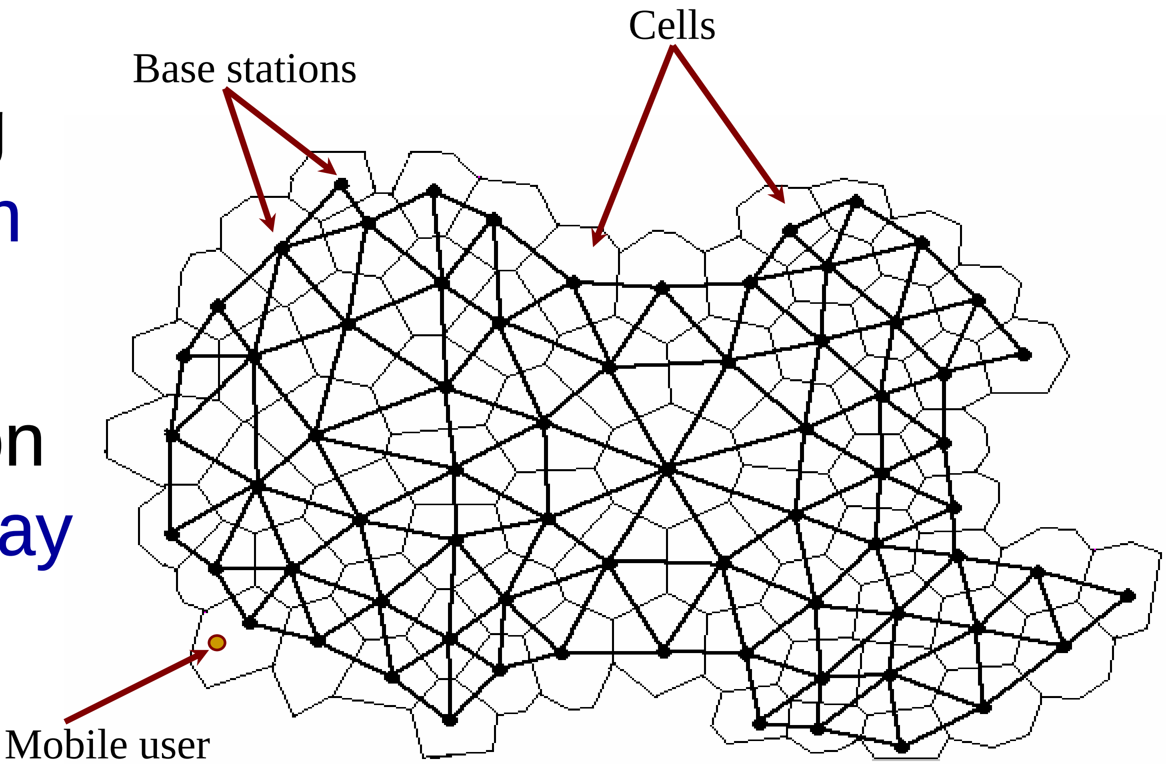


Our Irregular Cellular Networks

- We do not require from BSs to be set in a very regular pattern (➔ more flexibility in designing)

- Cells formed using the Voronoi diagram of BSs

- The communication graph is the Delaunay triangulation

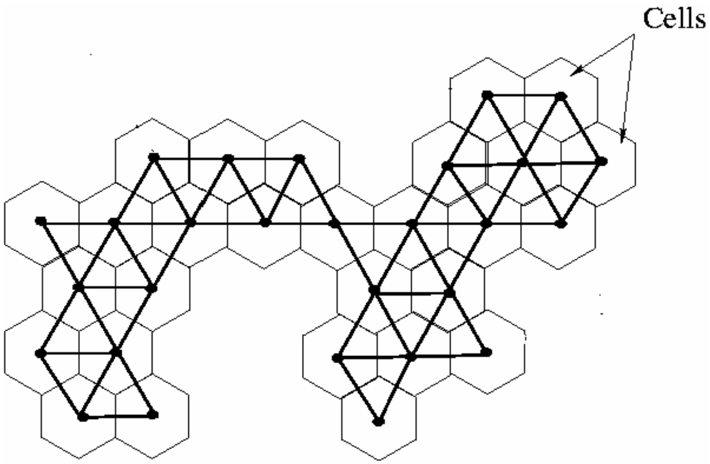


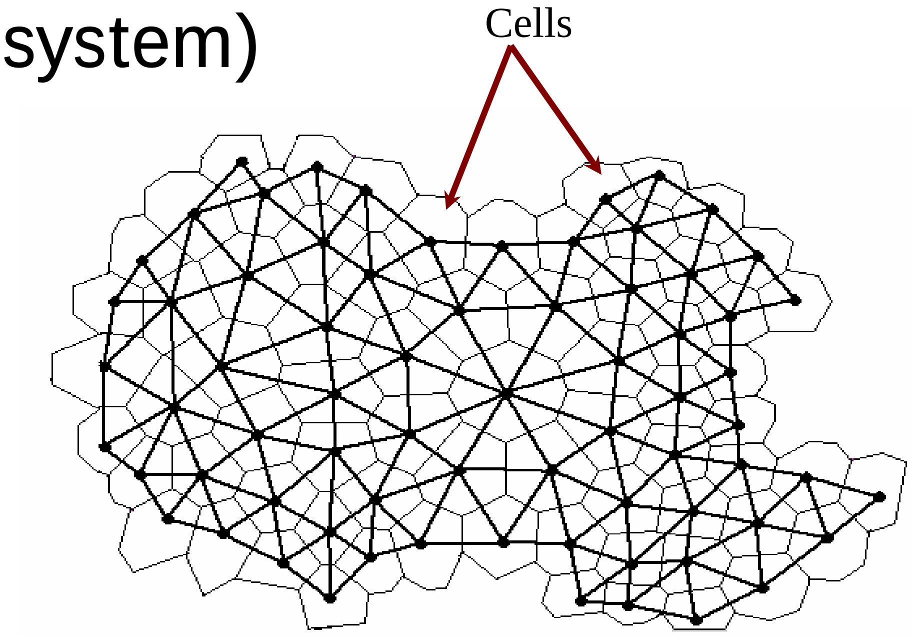
- Our only requirement: each inner cell has at least six neighbor cells (=6 in regular cellular networks)

Trigraphs

- If in the Voronoi diagram of BSs each inner cell has at least six neighbor cells (=6 in regular cellular networks)

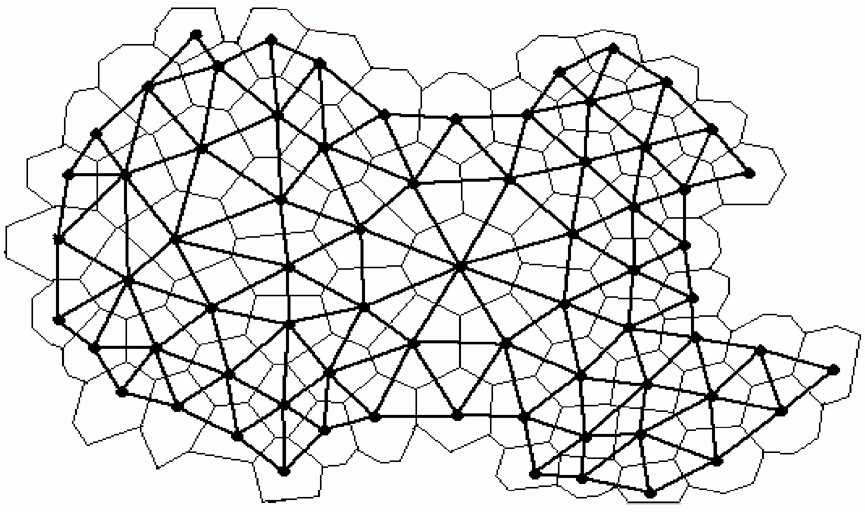
- ➔ (the Delaunay graphs=) Trigraphs are planar triangulations with inner vertices of degree at least six (if all =6 ➔ triangular system)



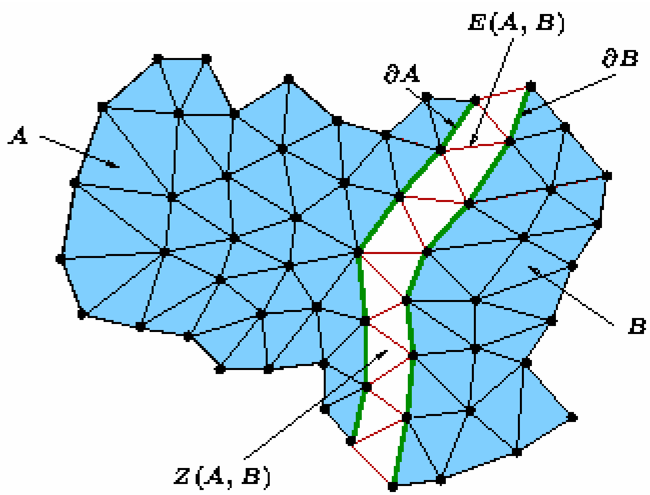


Our results for trigraphs

- Low depth hierarchical decomposition of a trigraph
  - ➔ distance labeling scheme with labels of size $O(\log^2 n)$ - *bits per node and constant time distance decoder*
  - ➔ routing labeling scheme with labels of size $O(\log^2 n)$ - *bits per node and constant time routing decision.*



Cuts in Trigraphs

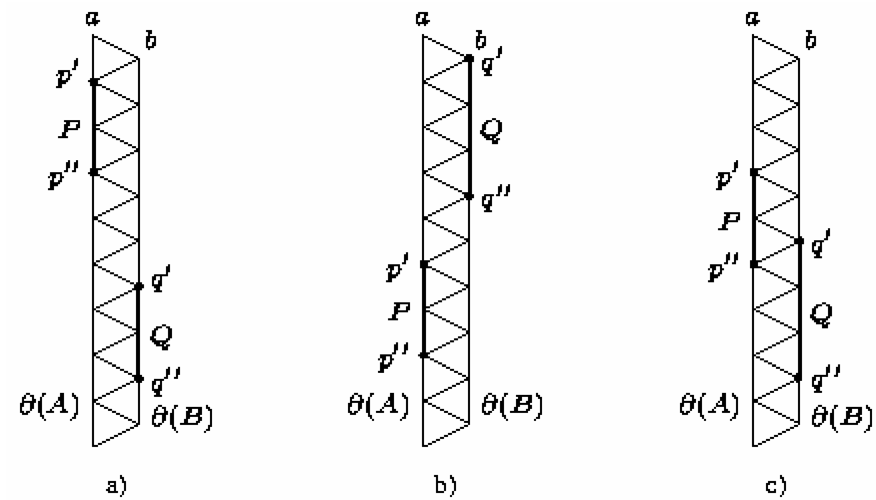


A *convex* cut $\{A, B\}$, its zone $Z(A, B)$, its border lines ∂A and ∂B and the set $E(A, B)$ of edges crossed by this cut.

- Border lines are shortest paths
- A and B parts are convex
- Projections are subpaths

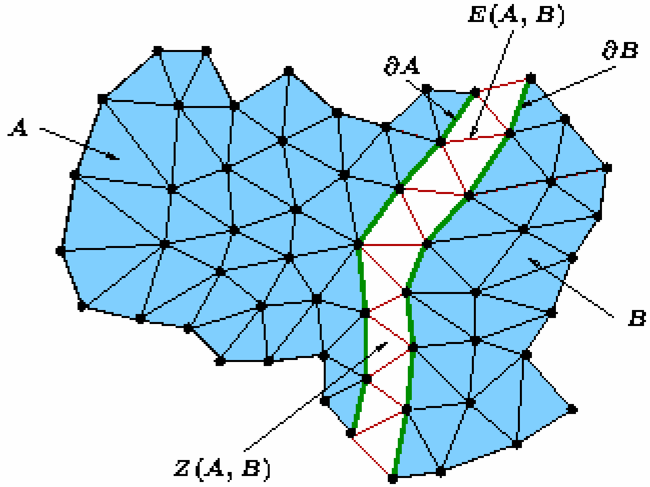
- Distance formula

$$d(x, y) = d(x, P) + d(P, Q) + d(y, Q)$$

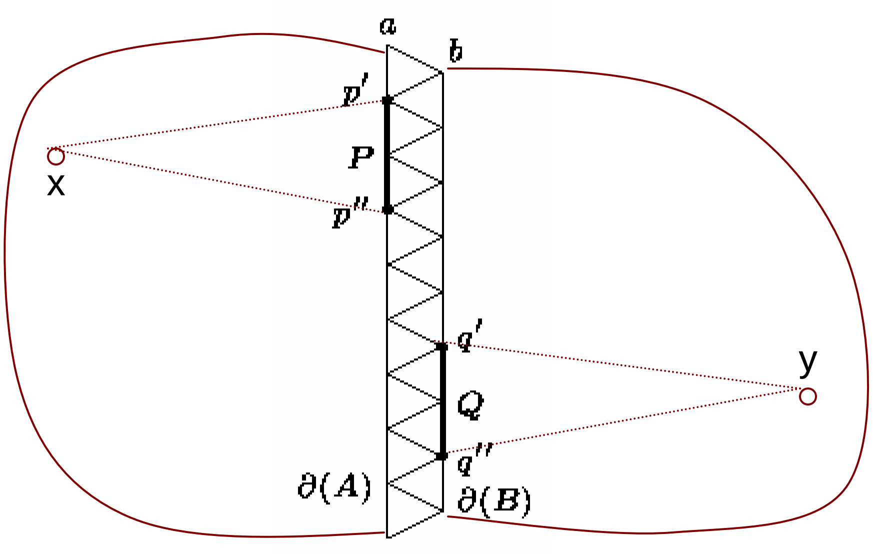


a) $d(P,Q) = d(p'', q') = d(b, q') - d(a, p'') + 1$; **b)** $d(P,Q) = d(p', q'') = d(a, p') - d(b, q'')$; **c)** $d(P,Q) = 1$.

Distances via cut



A *convex* **cut** $\{A, B\}$**, its zone** $Z(A, B)$**, its border lines** ∂A **and** ∂B **and the set** $E(A, B)$ **of edges crossed by this cut.**



- Distance formula

$$d(x, y) = d(x, P) + d(P, Q) + d(y, Q)$$

- Necessary information

$$D_x := \overset{1}{(1,} \quad \overset{2}{d(x, P),} \quad \overset{3}{d(p', a),} \quad \overset{4}{d(p'', a))}$$

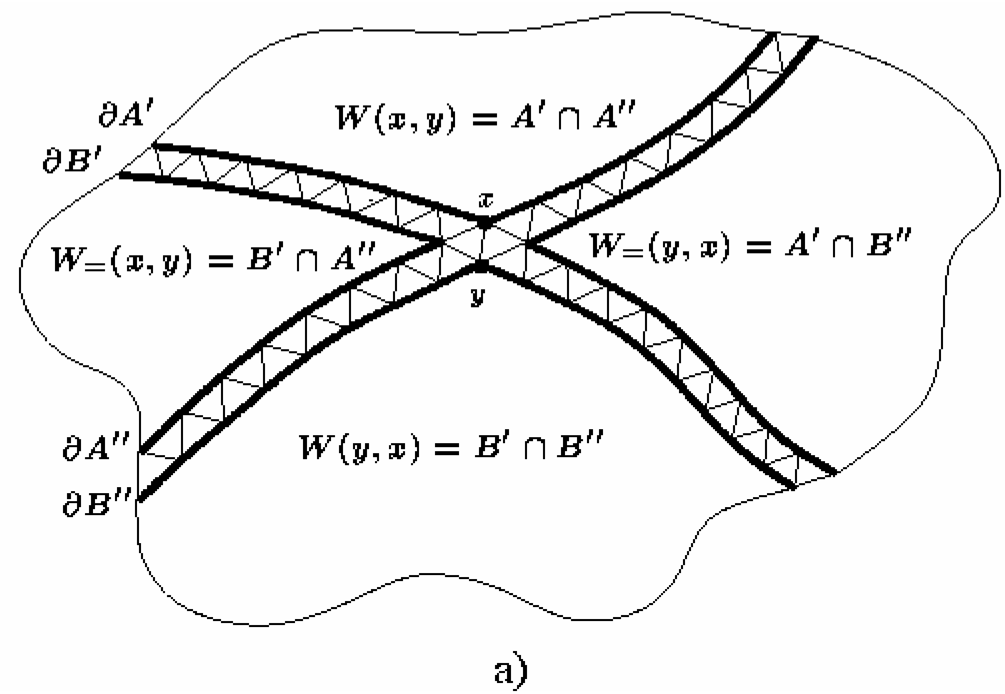
$$D_y := \overset{1}{(0,} \quad \overset{2}{d(y, Q),} \quad \overset{3}{d(q', b),} \quad \overset{4}{d(q'', b))}$$

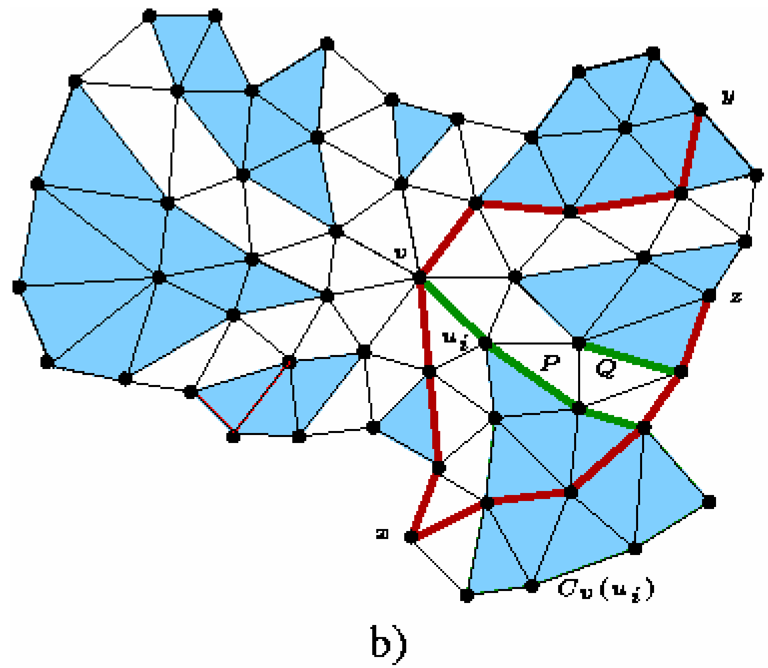
- Decoder

function distance_graphs(D_x, D_y)

if $D_x(1) = 0$ then /* rename inputs */ set
$C := D_x, D_x := D_y, D_y := C$
if $D_x(4) \leq D_y(3)$ then
return $D_x(2) + (D_y(3) - D_x(4) + 1) + D_y(2)$
else if $D_x(3) > D_y(4)$ then
return $D_x(2) + (D_x(3) - D_y(4)) + D_y(2)$
else return $D_x(2) + 1 + D_y(2)$

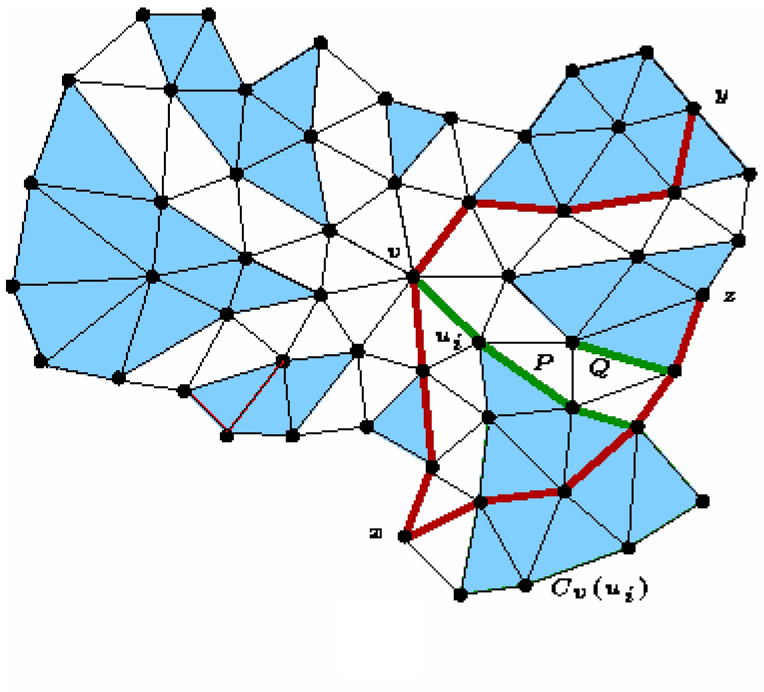
Decomposition: partition into cones

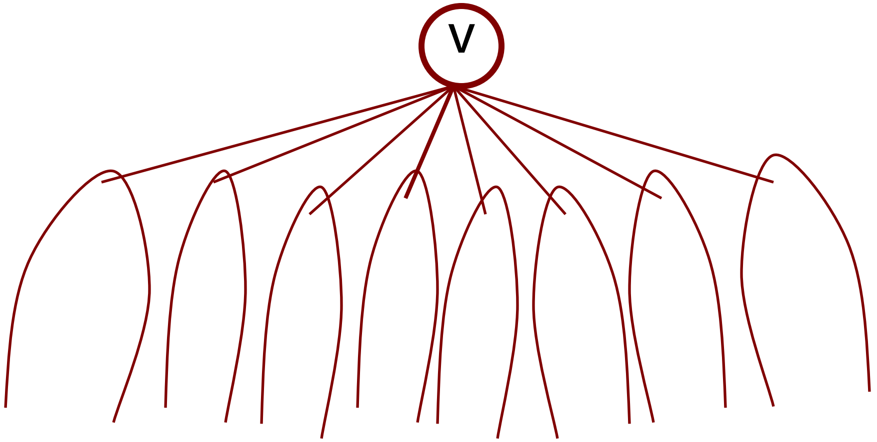




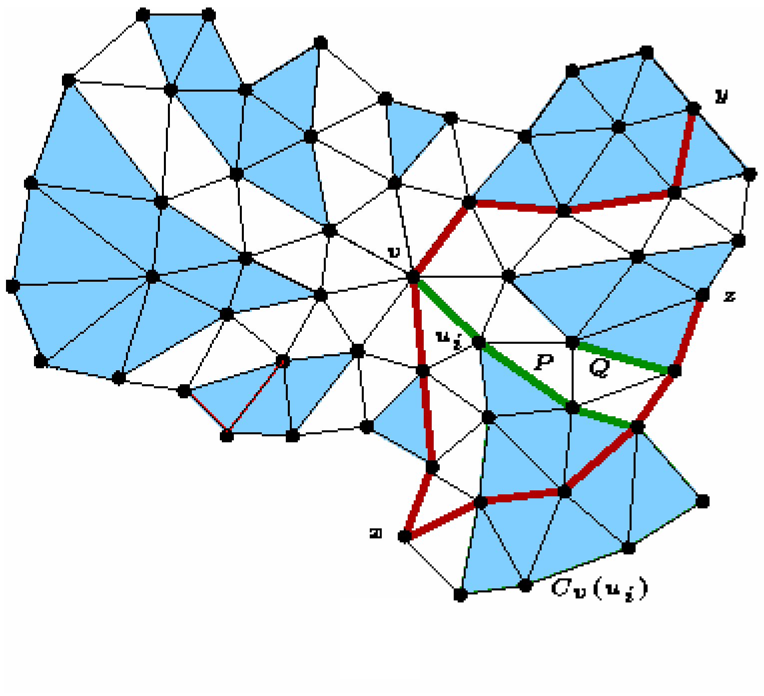
a) The pair of alternating cuts crossing the edge $e = xy$; b) The partition into cones around the vertex v. Since x and z lie in 2-neighboring cones, we have $d(x, z) = d(x, P) + d(P, Q) + d(Q, z)$. On the other hand, x and y lie in 3-neighboring cones implying $d(x, y) = d(x, v) + d(v, y)$.

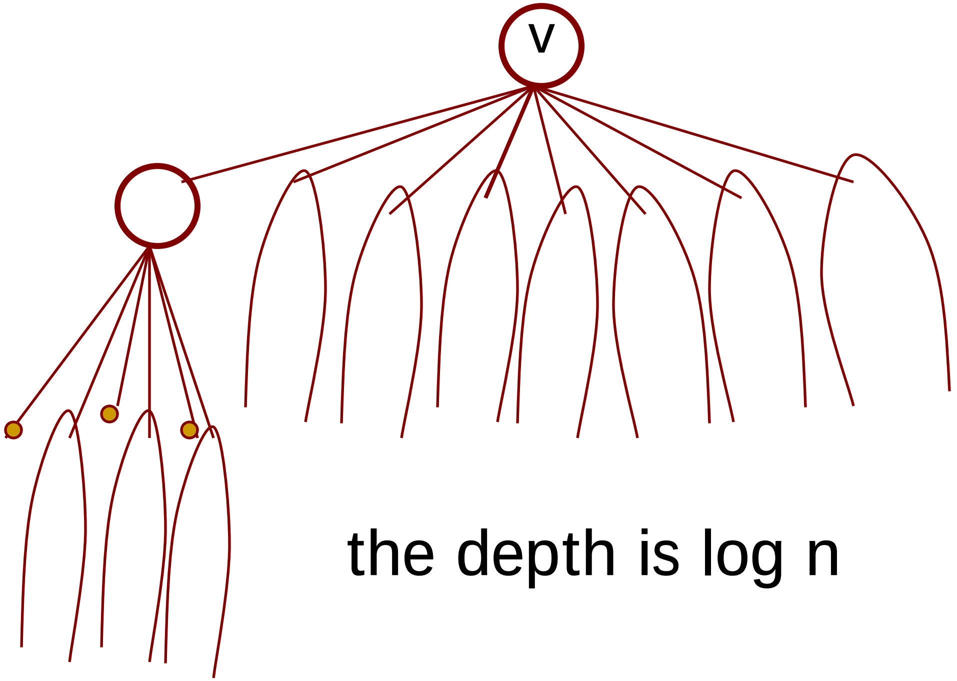
Decomposition tree





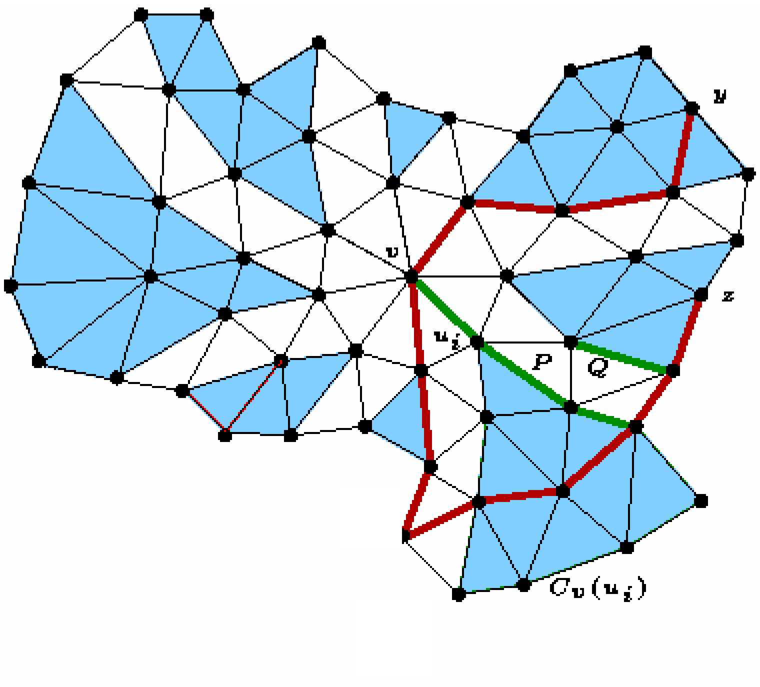
Decomposition tree

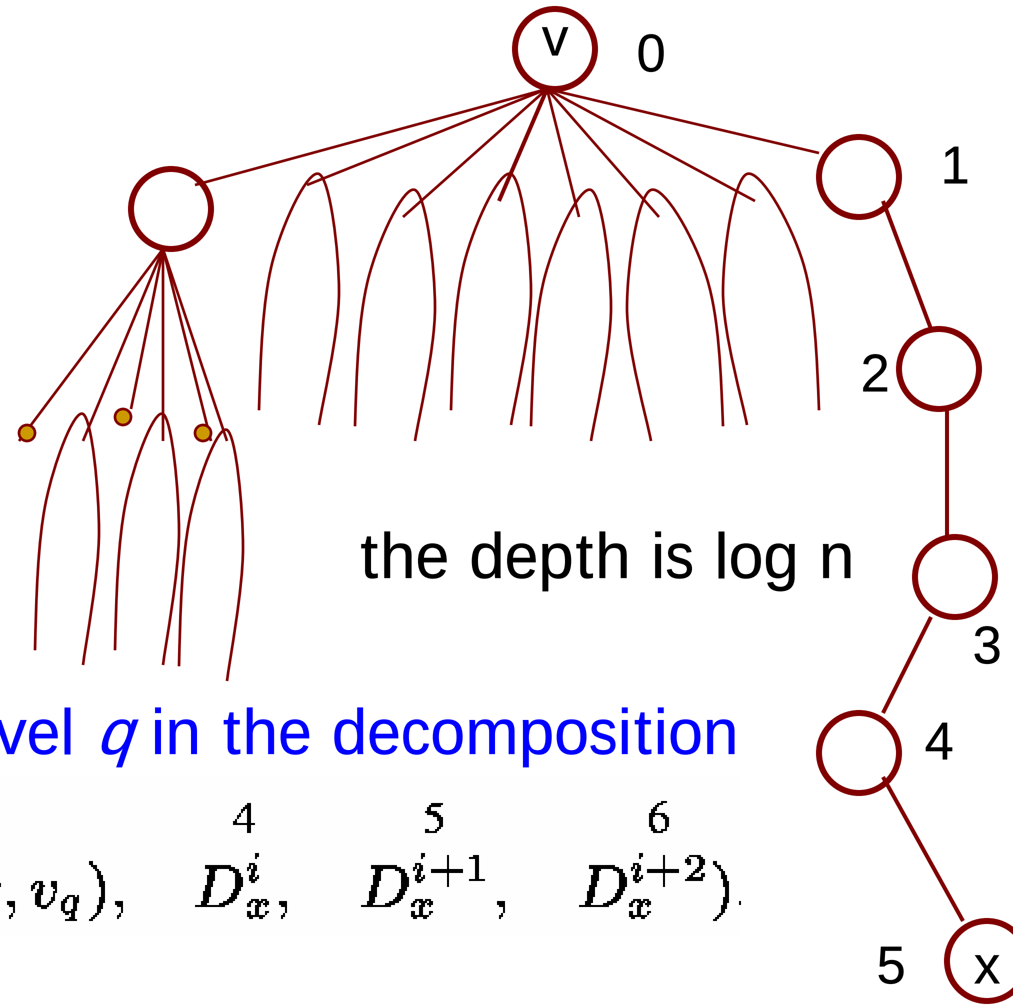




the depth is log n

The decomposition tree and the labels





- Necessary information for level *q* in the decomposition

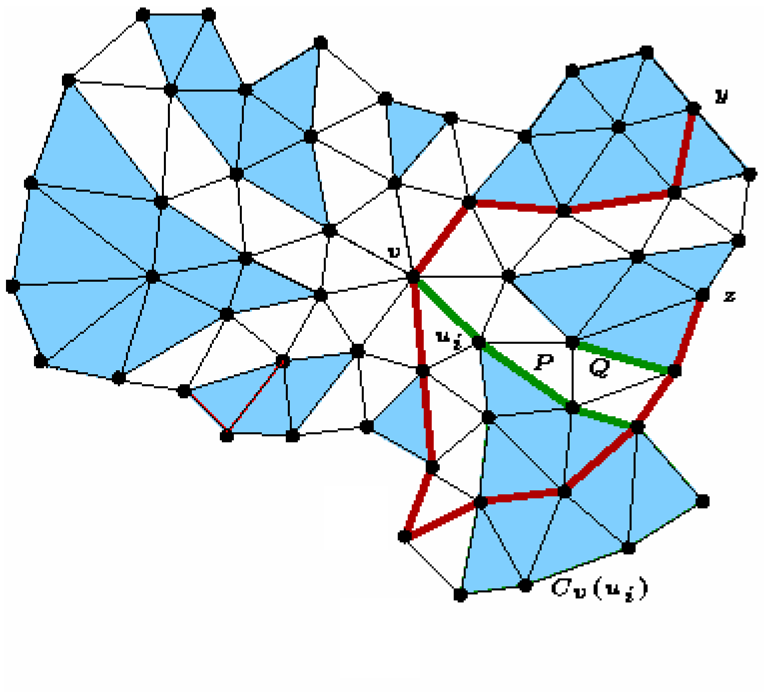
$$\tau_q^x := \overset{1}{(i,}\ \overset{2}{\delta_{G_q}(v_q),}\ \overset{3}{d_{G_q}(x, v_q),}\ \overset{4}{D_x^i,}\ \overset{5}{D_x^{i+1},}\ \overset{6}{D_x^{i+2})}$$

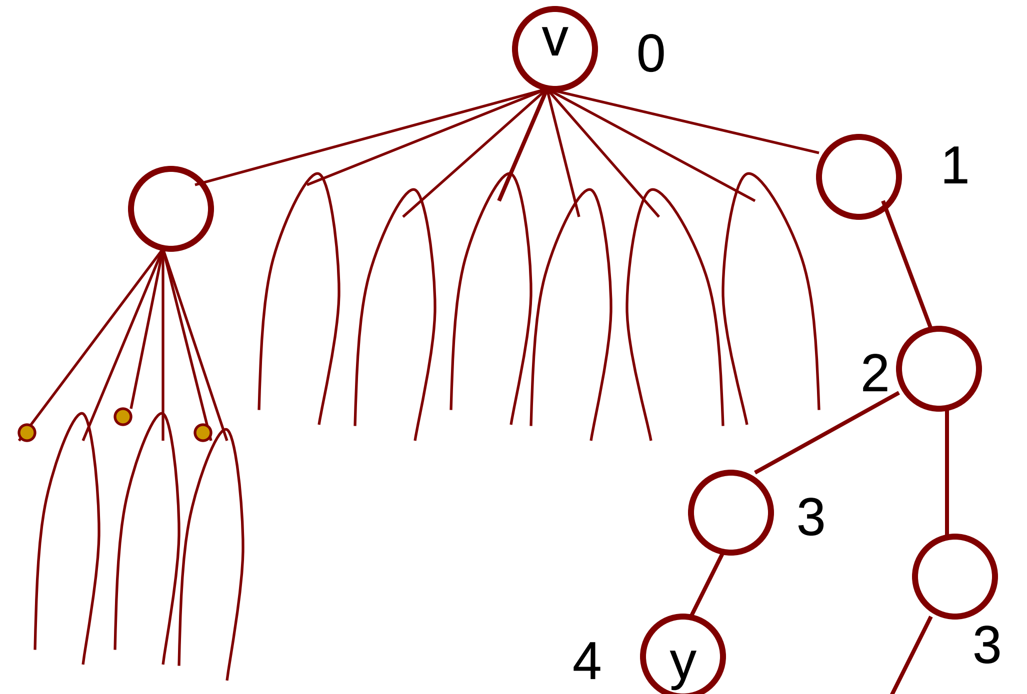
- The Labels

$$L(x) = A_x \circ \tau_0^x \circ \tau_1^x \circ \cdots \circ \tau_h^x$$

($O(\log^2 n)$ bits)

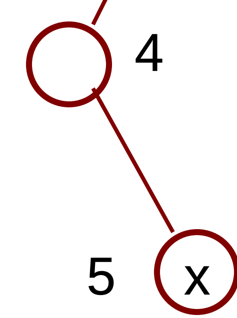
The decomposition tree and the labels





- Necessary information for level *q* in the decomposition

$$\tau_q^x := \overset{1}{(i,}\ \overset{2}{\delta_{G_q}(v_q),}\ \overset{3}{d_{G_q}(x, v_q),}\ \overset{4}{D_x^i,}\ \overset{5}{D_x^{i+1},}\ \overset{6}{D_x^{i+2})}.$$



- The Labels

$$L(x) = A_x \circ \tau_0^x \circ \tau_1^x \circ \cdots \circ \tau_h^x$$

($O(\log^2 n)$ bits)

$$L(y) = A_y \circ \tau_0^y \circ \tau_1^y \circ \cdots \circ \tau_q^y$$

Distance decoder

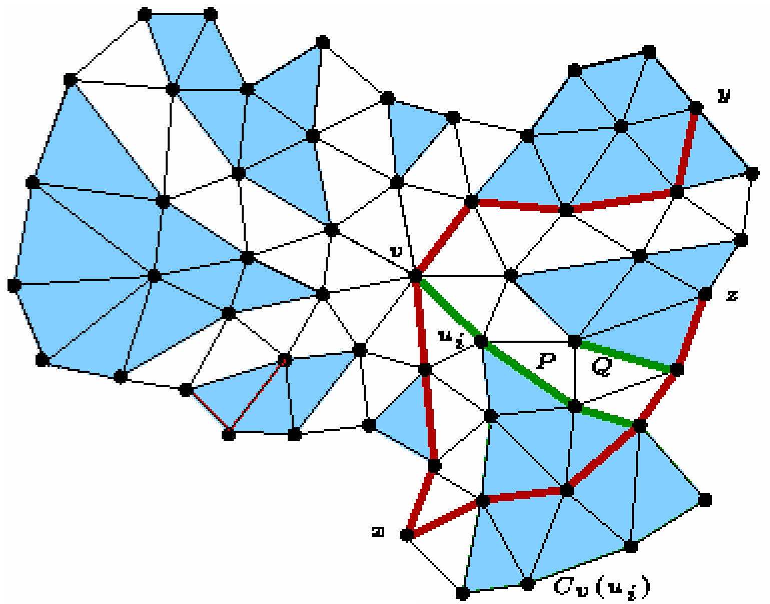
Algorithm DISTANCE_DECODER: Distance decoder for tri-graphs.

Input: two labels $L(x) = A_x \circ \tau_0^x \circ \tau_1^x \circ \cdots \circ \tau_h^x$ and $L(y) = A_y \circ \tau_0^y \circ \tau_1^y \circ \cdots \circ \tau_q^y$.

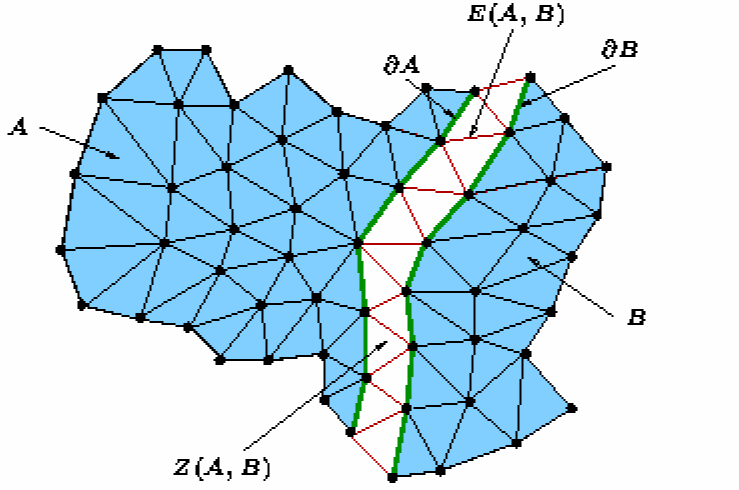
Output: $d(x, y)$, the distance between x and y in G.

Method:

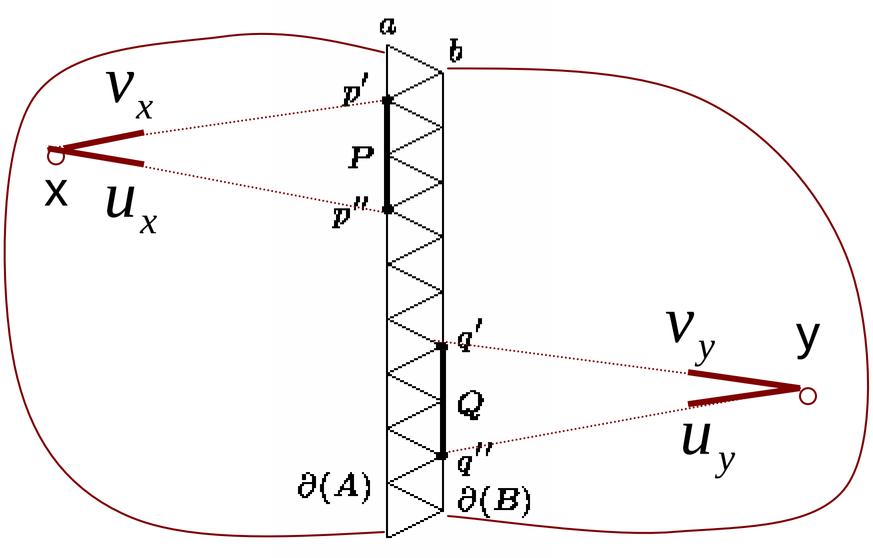
use A_x and A_y to find the depth l in $T(G)$ of
  the nearest common ancestor of $S(x)$ and $S(y)$;
extract from $L(x)$ and $L(y)$ the tuples τ_l^x and τ_l^y;
if $\tau_l^x(1) = \tau_l^x(2)$ then output $\tau_l^y(3)$ and stop; /* $x = v_q$ */
if $\tau_l^y(1) = \tau_l^y(2)$ then output $\tau_l^x(3)$ and stop; /* $y = v_q$ */
/* if the cones are 1-neighboring */
if $\tau_l^x(1) = \tau_l^y(1) - 1$ or $\tau_l^y(1) = 0$ and $\tau_l^x(1) = \tau_l^x(2) - 1$
  then output distance_graphs($\tau_l^x(5), \tau_l^y(4)$) and stop;
if $\tau_l^y(1) = \tau_l^x(1) - 1$ or $\tau_l^x(1) = 0$ and $\tau_l^y(1) = \tau_l^x(2) - 1$
  then output distance_graphs($\tau_l^y(5), \tau_l^x(4)$) and stop;
/* if the cones are 2-neighboring */
if ($\tau_l^x(1) = \tau_l^y(1) - 2$ or $\tau_l^y(1) = 0$ and $\tau_l^x(1) = \tau_l^x(2) - 2$
  or $\tau_l^y(1) = 1$ and $\tau_l^x(1) = \tau_l^x(2) - 1$) then output
  distance_graphs($\tau_l^x(6), \tau_l^y(4)$) and stop;
if ($\tau_l^y(1) = \tau_l^x(1) - 2$ or $\tau_l^x(1) = 0$ and $\tau_l^y(1) = \tau_l^x(2) - 2$
  or $\tau_l^x(1) = 1$ and $\tau_l^y(1) = \tau_l^x(2) - 1$) then output
  distance_graphs($\tau_l^y(6), \tau_l^x(4)$) and stop;
else output $\tau_l^x(3) + \tau_l^y(3)$.



Routing via cut



A *convex* cut $\{A, B\}$, its zone $Z(A, B)$, its border lines ∂A and ∂B and the set $E(A, B)$ of edges crossed by this cut.



- Necessary routing information

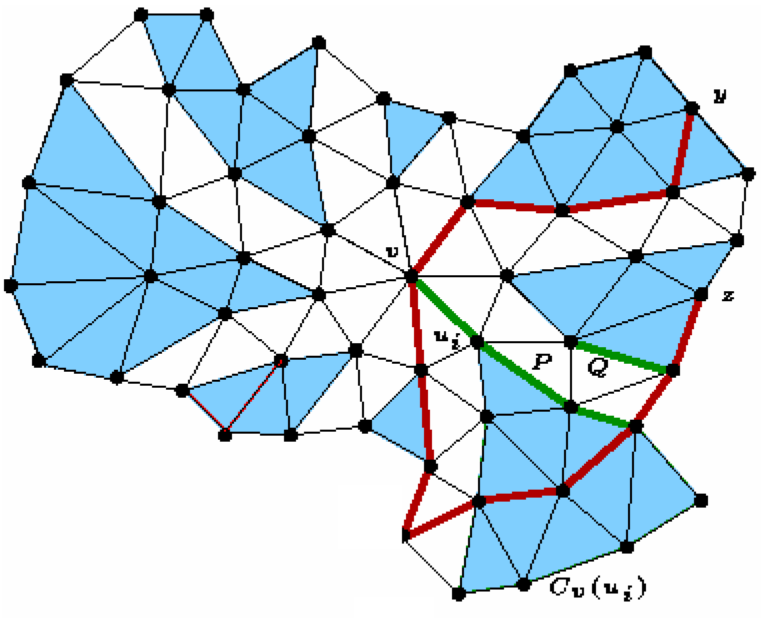
$$R_x := (\overset{1}{D_x}, \overset{2}{D_{v_x}}, \overset{3}{D_{u_x}}, \overset{4}{port(x, v_x)}, \overset{5}{port(x, u_x)}, \overset{6}{help(v_x)}, \overset{7}{help(u_x)})$$

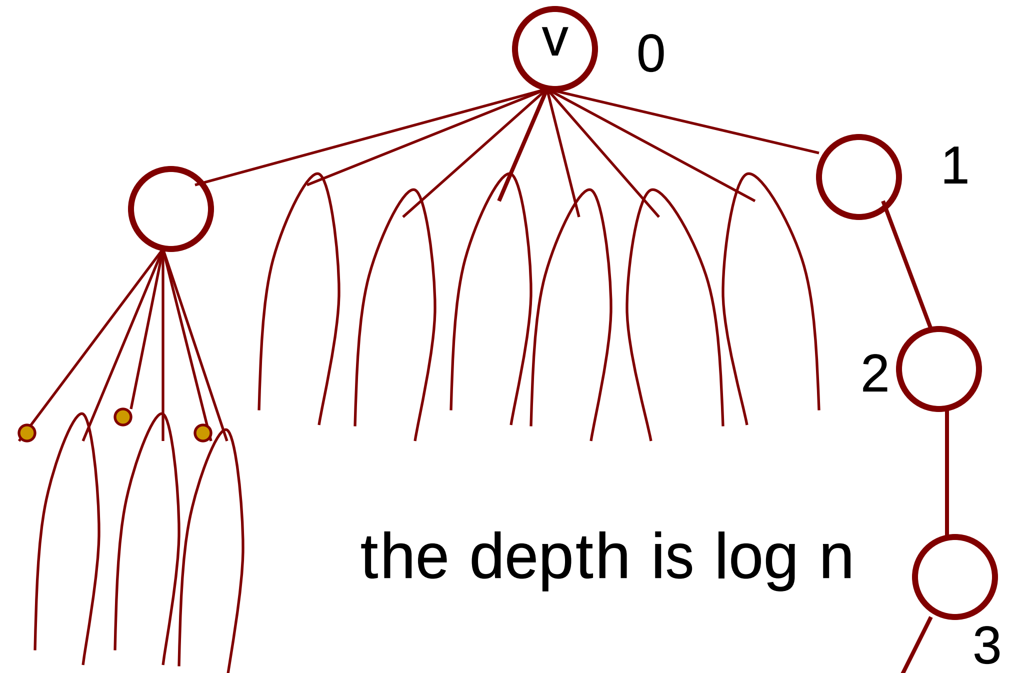
$$R_y := (\overset{1}{D_y}, \overset{2}{D_{v_y}}, \overset{3}{D_{u_y}}, \overset{4}{port(y, v_y)}, \overset{5}{port(y, u_y)}, \overset{6}{help(v_y)}, \overset{7}{help(u_y)})$$

- Decoder

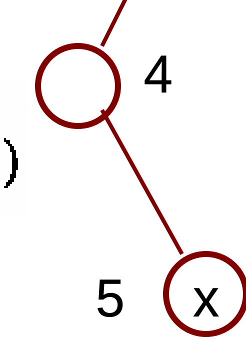
function routing_decision(R_x, R_y)
if $R_x(6) \neq 1$ then
  if distance_graphs($R_x(1), R_y(1)$) = distance_graphs($R_x(2), R_y(1)$) + 1 then output $R_x(4)$ else output $R_x(5)$
else if $R_x(7) \neq 1$ then
    if distance_graphs($R_x(1), R_y(1)$) = distance_graphs ($R_x(3), R_y(1)$) + 1 then output $R_x(5)$ else output $R_x(4)$
  else extract $D_y(4)$ from $R_y(1)$
    extract $D_{v_x}(3)$ from $R_x(2)$
    extract $D_{u_x}(3)$ from $R_x(3)$
    if $D_{u_x}(3) \leq D_{v_x}(3)$ then
      if $D_y(4) \leq D_{u_x}(3)$ then output $R_x(5)$
      else output $R_x(4)$
    else if $D_y(4) \leq D_{v_x}(3)$ then output $R_x(4)$
      else output $R_x(5)$

Routing labels





- Necessary information for level *q* in the decomposition



$$\mu_q^x := \overset{1}{(i,}\ \ \overset{2}{\delta_{G_q}(v_q)},\ \ \overset{3}{port_{G_q}(x, v_q)},\ \ \overset{4}{R_x^i},\ \ \overset{5}{R_x^{i+1}},\ \ \overset{6}{R_x^{i+2}},\ \ \overset{7}{port_{G_q}(v_q, x))}$$

- The Labels

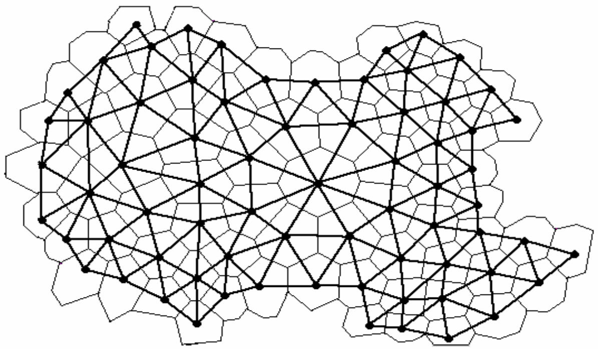
$$L(x) = A_x \circ \mu_0^x \circ \mu_1^x \circ \cdots \circ \mu_h^x$$

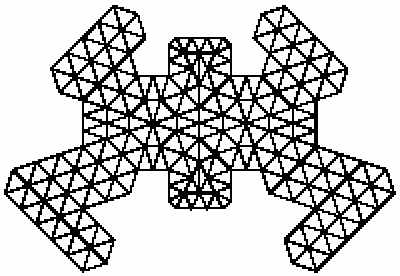
$$L(y) = A_y \circ \mu_0^y \circ \mu_1^y \circ \cdots \circ \mu_q^y$$

($O(\log^2 n)$ bits)

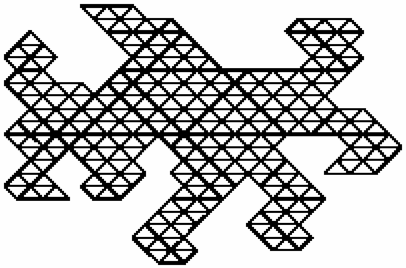
Main Result and Forthcomings

Thm: The family of n-vertex trigraphs enjoy distance and routing labeling schemes with $O(\log^2 n)$ -bit labels and constant time distance decoder and routing decision.

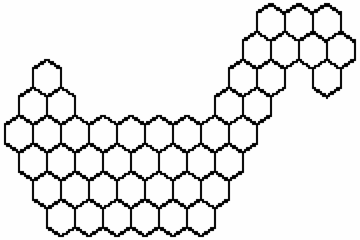




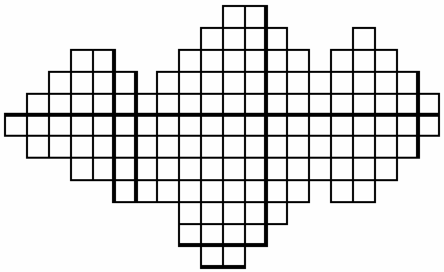
trigraph



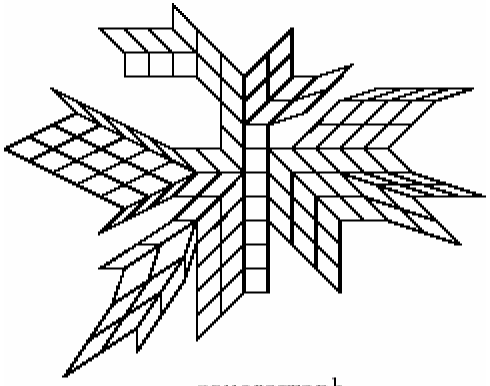
triangular system



hexagonal system



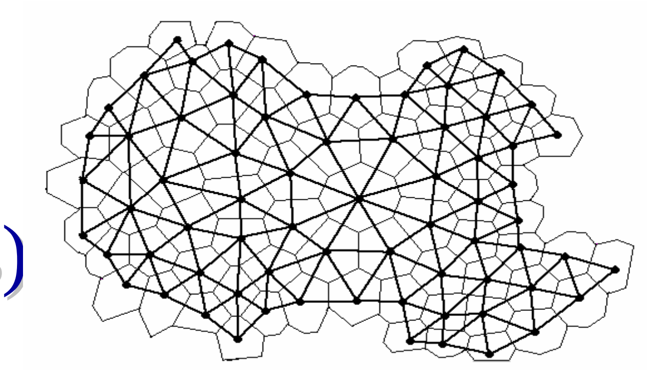
square system



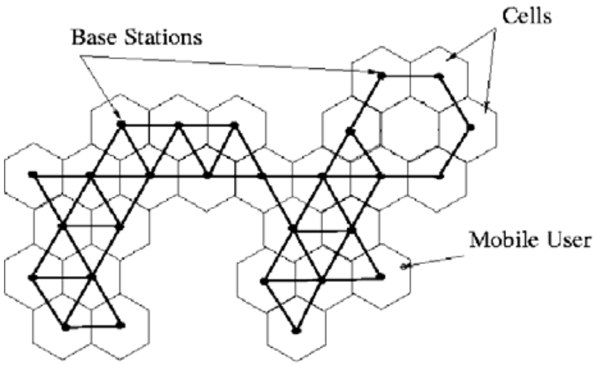
squaregraph

Open Problems

- Channel Assignment Problem in Irregular Cellular Networks ($L(p_1,\ldots,p_k)-$coloring in Trigraphs)



- BSs Placement Problem (resulting in a Trigraph)
  - Service area with demands, obstacles
  - Deploy min. # of BSs to cover area



- Not-Simply Connected Regular Cellular Networks (with holes)

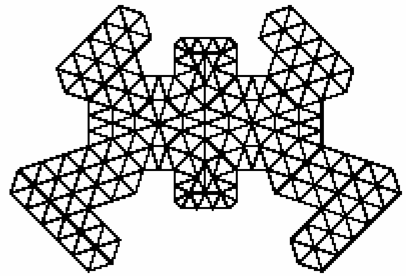
Thank you

Other Results

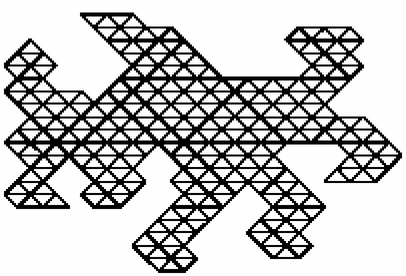
Thm: The families of *n*-node (6,3)-,(4,4)-,(3,6)-planar graphs enjoy distance and routing labeling schemes with $O(\log^2 n)$ -bit labels and constant time distance decoder and routing decision.

(p,q)-planar graphs:

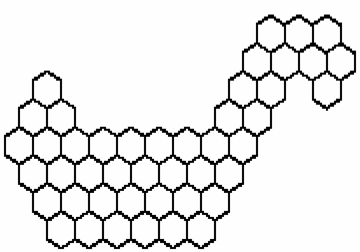
- inner faces of length at least p
- inner vertices of degree at least q



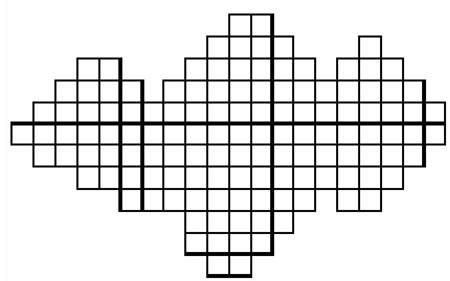
trigraph



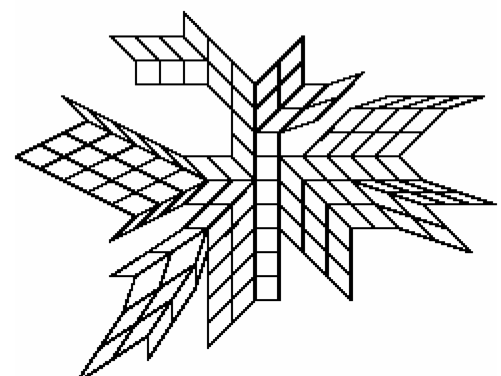
triangular system



hexagonal system



square system



squaregraph