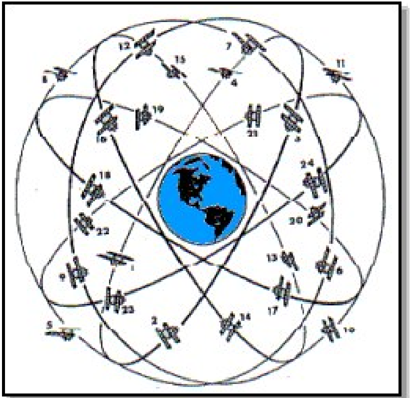
Communication network for the GPS III system

Simon Crevals
Prof. dr. G. Brinkmann
N. Van Cleemput

Department of Applied Mathematics and Computer Science
Ghent University

July 2010

1. Introduction
2. First test
3. Final program
4. Results
5. Conclusion




Advantages of communication

- continuous telemetry
- frequent updates

Restrictions for connections

- connection possible during the entire orbit
- 4 connections a satellite

Definition

V_{GPS} is the set of all satellites.

UNIVERSITEIT GENT

Definition

V_{GPS} is the set of all satellites.

Definition

$G_{Pos} = (V_{GPS}, E_{Pos})$, *with* $\forall v, w \in V_{GPS} : \{v, w\} \in E_{Pos} \Leftrightarrow$ *communication between satellites* v *and* w *is technically possible.*

Definition

V_{GPS} is the set of all satellites.

Definition

$G_{Pos} = (V_{GPS}, E_{Pos})$, with $\forall v, w \in V_{GPS} : \{v, w\} \in E_{Pos} \Leftrightarrow$ communication between satellites v and w is technically possible.

Definition

*A **connection graph** is a spanning, 4-regular subgraph of G_{Pos}.*

Definition

Given a graph $G = (V, E)$ *and a vertex* $v \in V$: *the* **neighbourhood** *of* v *in* G *is* $N(v, G) = \{w \in V | \{v, w\} \in E\}$.

Definition

Given a graph $G = (V, E)$ and a vertex $v \in V$: the **neighbourhood** *of v in G is $N(v, G) = \{w \in V | \{v, w\} \in E\}$.*

Definition

The function **uredop** *of a vertex v with respect to its neighbours is defined as:*

Let $M_4 = \{M \subseteq V_{GPS} || M| = 4\}$.

$uredop : V_{GPS} \times M_4 \rightarrow \mathbb{R}^+ \bigcup \{\infty\}$, so that

$\forall v \in V_{GPS} : M \not\subset N(v, G_{Pos}) \Rightarrow uredop(v, M) = \infty$.

UNIVERSITEIT GENT

Definition

The uredop value of a vertex v in a connection graph G is defined as:
$uredop(v, G) = uredop(v, N(v, G))$.

Definition

The uredop value of a vertex v in a connection graph G is defined as: $uredop(v, G) = uredop(v, N(v, G))$.

Definition

The uredop value of a connection graph G is defined as: $uredop(G) = max\{uredop(v, G) | v \in V_{GPS}\}$.

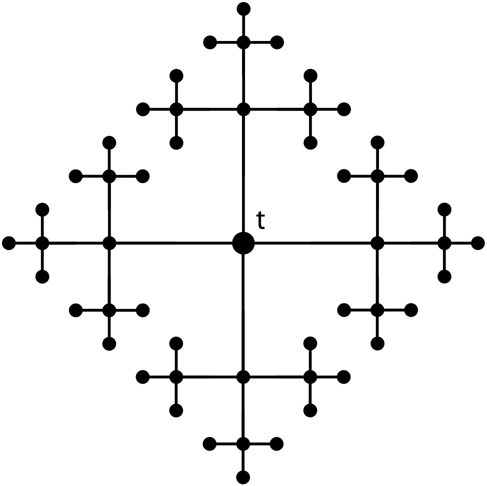
Minimum requirements

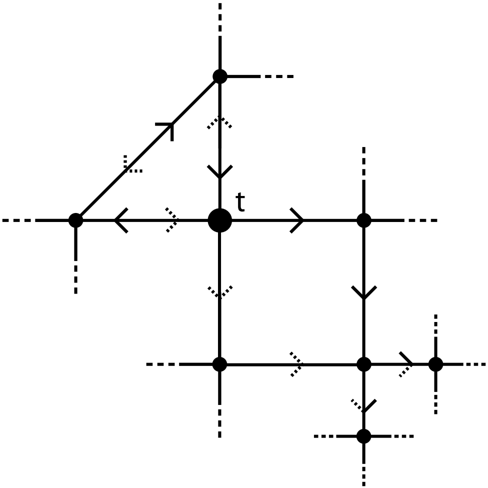
Construct a connection graph G which

- is 4-regular
- is a subgraph of G_{Pos}
- has diameter at most 4
- has $uredop(G) < 3$

Best connection graph G

- Diameter 3
- Smallest possible value for $uredop(G)$
- Maximum diameter 4 after one edge removal


t


t

Questions?

1. Introduction
2. First test
3. Final program
4. Results
5. Conclusion

Questions

- Are there connection graphs with diameter 3?
- How many?
- With which uredop value?

Method

- Generate 4-regular graphs
- Filter graphs with diameter 3
- Determine whether they are subgraph of G_{Pos}
- Determine uredop value

Results

- Thousands of millions of 4-regular graphs with 27 vertices and diameter 3
- Almost all tested graphs were subgraph of G_{Pos}
- Very different uredop values (also good ones)

Results

- Thousands of millions of 4-regular graphs with 27 vertices and diameter 3
- Almost all tested graphs were subgraph of G_{Pos}
- Very different uredop values (also good ones)

Conclusion

The uredop values will have to provide the biggest restriction.

Definition

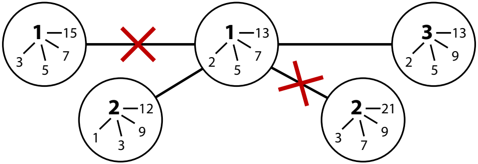
Given a graph $G = (V, E)$*. A vertex set* $IS \subseteq V$ *is an* **independent set** *in* $G \Leftrightarrow \forall v, w \in IS : \{v, w\} \notin E$.

Definition

Given a graph $G = (V, E)$*. A vertex set* $C \subseteq V$ *is a* **clique** *in* $G \Leftrightarrow \forall v, w \in C : \{v, w\} \in E$.
If there doesn't exist a clique C' *with* $|C'| > |C|$*, then* C *is a maximum clique in* G.

Definition

$G_C(u) = (V_C(u), E_C(u))$, *with*
$V_C(u) = \{(s, N) | s \in V_{GPS}, N \subset N(s, G_{Pos}), |N| = 4$ *and*
$uredop(s, N) < u\}$ *and*
$E_C(u) = \{\{v, w\} | v = (s, N_s) \in V_C(u), w = (t, N_t) \in V_C(u), s \neq t$ *and*
$s \in N_t \Leftrightarrow t \in N_s\}$.



Definition

$G_C(u) = (V_C(u), E_C(u))$, *with*
$V_C(u) = \{(s, N) | s \in V_{GPS}, N \subset N(s, G_{Pos}), |N| = 4$ *and*
$uredop(s, N) < u\}$ *and*
$E_C(u) = \{\{v, w\} | v = (s, N_s) \in V_C(u), w = (t, N_t) \in V_C(u), s \neq t$ *and*
$s \in N_t \Leftrightarrow t \in N_s\}$.

Theorem

The largest cliques in $G_C(u)$ have at most 27 vertices.

Definition

$G_C(u) = (V_C(u), E_C(u))$, *with*
$V_C(u) = \{(s, N) | s \in V_{GPS}, N \subset N(s, G_{Pos}), |N| = 4$ *and*
uredop$(s, N) < u\}$ *and*
$E_C(u) = \{\{v, w\} | v = (s, N_s) \in V_C(u), w = (t, N_t) \in V_C(u), s \neq t$ *and*
$s \in N_t \Leftrightarrow t \in N_s\}$.

Theorem

The largest cliques in $G_C(u)$ have at most 27 vertices.

Theorem

Every connection graph $G = (V_{GPS}, E)$ with uredop values smaller than u, corresponds to a maximum clique C in $G_C(u)$, with $|C| = 27$.

Definition

$G_C(u) = (V_C(u), E_C(u))$, with
$V_C(u) = \{(s, N) | s \in V_{GPS}, N \subset N(s, G_{Pos}), |N| = 4$ and
$uredop(s, N) < u\}$ and
$E_C(u) = \{\{v, w\} | v = (s, N_s) \in V_C(u), w = (t, N_t) \in V_C(u), s \neq t$ and
$s \in N_t \Leftrightarrow t \in N_s\}$.

Theorem

Every maximum clique C in $G_C(u)$, with $|C| = 27$, corresponds to a connection graph with uredop smaller than u.

UNIVERSITEIT GENT

Definition

The language **Uredop** *is the set of all strings with the structure* $g'\#c\#u$, *with*

- g' *represents a graph* G'
- c *represents all possible sets of 4 neighbours with corresponding value for each vertex*
- u *represents a natural number* U

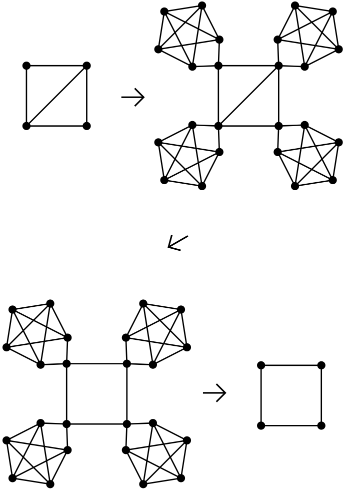
and for which there exists a subgraph G *of* G', *such that* $C(G) < U$.

Theorem

The language Uredop is NP-complete.

Proof

It is easy to see that Uredop $\in$ *NP.*
We still have to prove that each problem in NP can be reduced to Uredop in polynomial time.
To prove this, we reduce the language HC (graphs containing a Hamiltonian cycle) to Uredop.



General clique solver

Too slow for this specific problem

Pseudocode recursion

- Determine central vertex with smallest number of possible neighbourhoods
- Amount = 0: backtrack
- For each possible neighbourhood:
  - choose neighbourhood
  - adjust lists with possible neighbourhoods
  - found connection graph or continue in recursion

Some details

- Diameter as bounding criterium
- Use of bitvectors
- Reduce iteration length
- Efficient adjustment of lists with possible neighbourhoods

Adjust list with possible neighbourhoods

- Split lists once every recursion step
- Only adjust lists of (not yet visited) neighbours G_{Pos}
- Pass the corresponding lists

UNIVERSITEIT GENT

Adjust list with possible neighbourhoods

- Split lists once every recursion step
- Only adjust lists of (not yet visited) neighbours G_{Pos}
- Pass the corresponding lists

Split with and without 2

1	2	5	9	13
1	4	7	13	19
1	0	2	7	19
1	0	3	5	9
1	2	3	4	13

Adjust list with possible neighbourhoods

- Split lists once every recursion step
- Only adjust lists of (not yet visited) neighbours G_{Pos}
- Pass the corresponding lists

Split with and without 2

1	2	5	9	13
1	2	3	4	13
1	0	2	7	19
1	0	3	5	9
1	4	7	13	19

Programs

- Independent set method
- Independent set method (Gunnar Brinkmann)
- General clique solver (Patric Östergård)

Determine correctness

- Compare some specific graphs
- Compare the number of graphs per diameter, given an upper bound for uredop

Best graphs

Restrictions	Diameter (D)	Diameter after edge removal (E)	uredop value
D 4	4	5	0.92423
D 4, E 4	4	4	0.93317
D 3	3	5	0.96598
D 3, E 4	3	4	0.97854

For our practical problem

- Best graphs within the hour
- Much better results than those found by Lockheed Martin

Time needed depends heavily on the input weights.

Possible variations of input:

- Different amount of vertices
- Different values weight function
- Different amount of connections per vertex
- Different G_{Pos}

Thank you!