Building Highly Reliable Networks with GRASP/VND Heuristics

Mathias Bourel, Eduardo Canale, Franco Robledo, Pablo Romero, Luis Stábile

15^{th} International Conference on
Design of Reliable Communication Networks.

DRCN, 2019

Motivation

Remark

- *Reliability falls within the field of metrology.*
- *Its practical interests is found in network design.*
- *Choose links to maximize reliability & connectivity.*
- *Minimum-cost topologies are partially known (eg. kECON).*
- *What happens under probabilistic models?*

Main Goal

We are given $2r$ *terminals.*
Choose $3r$ *links to achieve maximum reliability.*

Problem

Definition (Unreliability)

The *unreliability* of a simple graph G with independent link failures with probability ρ is:

$$U_G(\rho) = \sum_{k=0}^{q} m_k \rho^k (1-\rho)^{q-k},$$

being m_k the number of ways to disconnect G removing k links. A (p, q)-graph is a graph with p nodes and q links.

Definition (Uniformly Most-Reliable Graph (UMR))

A (p, q)-graph G is UMR if its unreliability is minimum among all (p, q)-graphs and all $\rho \in [0, 1]$.

Necessary and Sufficient Conditions

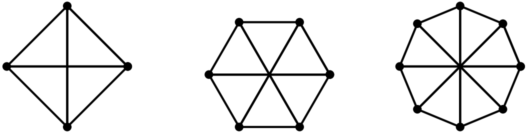
Proposition (Sufficient Condition)

If $m_k(G) \leq m_k(H)$ for all k and all (p, q)-graph H, then G is uniformly most-reliable.

Proposition (Necessary Condition)

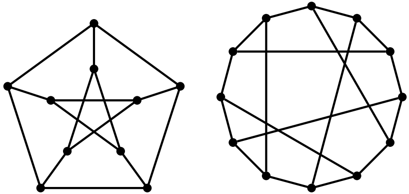
Optimal graphs G must have the maximum tree-number $\tau(G)$, maximum connectivity $\lambda(G)$, and the minimum number of disconnecting sets $m_\lambda(G)$.

Known Cubic UMR Graphs



(a) $K_4 = M_2$ (b) $K_{(3,3)} = M_3$ (c) M_4

Figure: Complete, Bipartite and Wagner graphs.



(d) P_5 (e) Y_6

Figure: Petersen and Yutsis graphs.

Heuristic - Main Idea

- Minimize each coefficient m_k in individual blocks.
- Combine them in a Variable Neighborhood Search (VND).
- The previous idea is computationally prohibitive!
- At least, force the necessary conditions.
- Use GRASP/VND to return a candidate.
- The objective should be only one...

Objective

Conjecture (Boesch et. al.)

If G is uniformly most-reliable (p, q)-graph, then $m_k(G) \leq m_k(H)$ for all (p, q)-graph H.

Remark

If Boesch conjecture holds, then the number of disconnected subgraphs $m(G) = \sum_{k=0}^{q} m_k$ must be minimized.

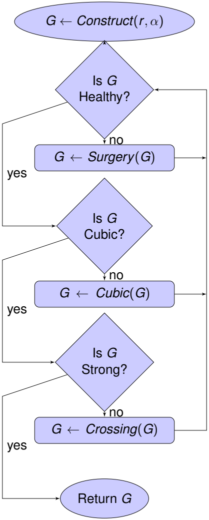
Finding $m(G)$ is $\mathcal{NP}$-Hard (an evaluation of Tutte polynomial).
Observe that $m(G) = 2^q \times U_G(1/2)$.
An estimation for $U_G(1/2)$ is available using Monte Carlo.

GRASP/VND

Algorithm 1 $G = HighlyReliable(r, iter, \alpha)$

$G \leftarrow M_r$
for $i = 1$ **to** *iter* **do**
    $G_{input} \leftarrow Construct(r, \alpha)$
    $G(i) \leftarrow VND(G_{input})$
    if $m(G(i)) \leq m(G)$ **for all** k **then**
        $G \leftarrow G(i)$
    end if
end for
return G

VND - Flow Diagram



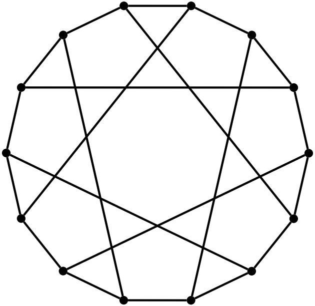
Experimental Setting

Settings

- $r \in \{7, \ldots, 15\}$.
- $iter = 10^6$ (executions of the GRASP/VND)
- $N = 10^4$ (sample graphs in Monte Carlo)
- Output: 9 graphs: R_r.
- Brute force test: they are the only UMR candidates.

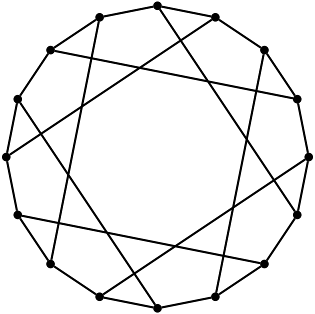
Results: R_7 and R_8

Heawood Graph



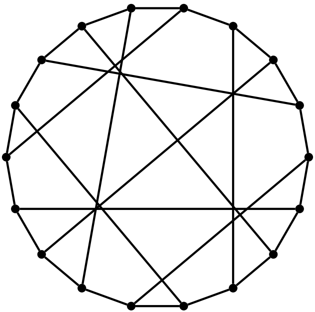
$U_{R_7}(\frac{1}{2}) = 0.9177$
$\tau(R_7) = 50421$

Möbius-Kantor Graph

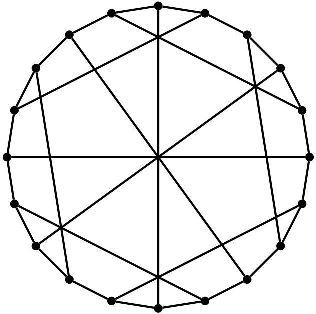


$U_{R_8}(\frac{1}{2}) = 0.94320$
$\tau(R_8) = 248832$

Results: R_9 and R_{10}

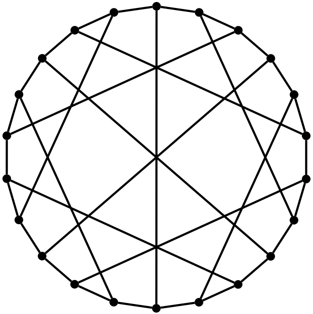


$U_{R_9}(\frac{1}{2}) = 0.9607$
$\tau(R_9) = 1265625$

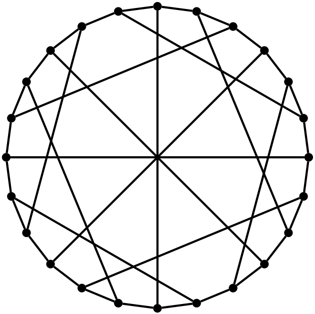


$U_{R_{10}}(\frac{1}{2}) = 0.97310$
$\tau(R_{10}) = 6422000$

Results: R_{11} and R_{12}

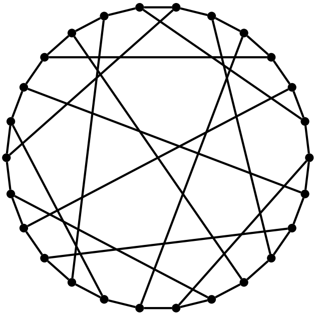


$U_{R_{11}}(\frac{1}{2}) = 0.9801$
$\tau(R_{11}) = 32710656$

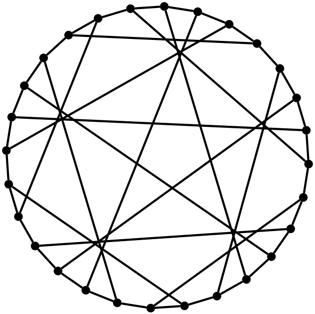


$U_{R_{12}}(\frac{1}{2}) = 0.988$
$\tau(R_{12}) = 168664320$

Results: R_{13} and R_{14}

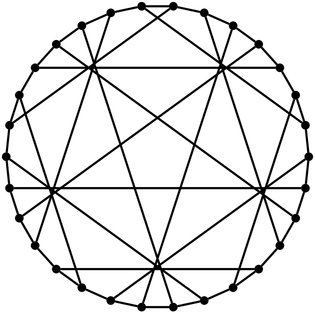


$U_{R_{13}}(\frac{1}{2}) = 0.9904$
$\tau(R_{13}) = 862488000$



$U_{R_{14}}(\frac{1}{2}) = 0.9931$
$\tau(R_{14}) = 4410450000$

Results: R_{15}



$$U_{R_{15}}(\tfrac{1}{2}) = 0.9956$$
$$\tau(R_{15}) = 23066015625$$

Conclusions

1. Finding UMR graphs is hard.
2. A methodology to find highly reliable networks is proposed.
3. New candidates of UMR graphs are also found.
4. UMR graphs are highly symmetrical.
5. Several conjectures are still open.
6. Only few works deal with node-failures.