# Graph Classification and Easy Reliability Polynomials*

**Pablo Romero and Gerardo Rubino**

**Abstract.** A graph classification is provided, inspired in an interplay between a celebrated network reliability problem and counting. This problem is precisely the probability that a random graph is connected following Gilbert edge-deletion rule. The graph classification induces a classification of reliability polynomials. Finally, a level of difficulty is assigned to graphs, and provides a corresponding notion of "easy reliability polynomials".

**1. INTRODUCTION.** Historically, several counting problems are related with graphs. They appear naturally in the practice of engineering, physics, chemistry, biology, economics, and other branches of knowledge. For instance, counting spanning trees helped to solve electrical circuits to Gustav Kirchhoff in the nineteenth century [**8**], and counting marriages in bipartite graphs launched a new complexity hierarchy in classes of counting problems, headed by Leslie Valiant [**10**]. Up to that moment, mathematicians were surprised with the fact that it is simple to find the determinant of a matrix but the permanent (in brief words, a permanent is a determinant but where all permutations have the same sign). Indeed, finding the permanent of a matrix is a hard task, and it was formalized by Leslie Valiant proving that it is the first problem inside a special "hard counting class", called #P-Complete class. Informally, this class captures the complexity of NP-Hard decision problems in counting.

In this paper, we want to play with a counting problem which captures an abstract setting from telecommunications. We are given a connected graph, where nodes are perfect but edges may fail with identical and independent probability $q = 1 - p \in [0, 1]$ (this is Gilbert model for random graphs, where $p$ is the probability of edge-presence [**6**]). The resulting graph may be either connected or not. We aim to find the probability that the random graph is connected, or connectedness probability, denoted by $R(p)$. At a first sight, it does not seem a counting problem. However, we will see this is nothing but a counting problem!

Indeed, let us consider a simple graph $G = (V, E)$ with $m = |E|$ edges and $n = |V|$ nodes. We will stick to this symbology during the treatment. Let us call $F_i$ to the number of subgraphs with precisely $m - i$ edges. They all have the same probability, to know, $p^{m-i}(1-p)^i$. Summing all events we get that:

$$R(p) = \sum_{i=0}^{m} F_i p^{m-i}(1-p)^i. \tag{1}$$

Therefore, finding $R(p)$ is equivalent to count all entries of the "$F$-vector" $F = (F_0, \ldots, F_m)$. This counting problem is called network reliability analysis problem, and it remains in the heart of network reliability theory [**1**]. Effort performed by Michael Ball and Scott Provan permitted to determine that counting the $F$-vector is a hard task again [**2**]. In this paper we want to answer two questions:

*This work has been partially supported by the Stic AmSud project "AMMA" 2013-2014.

(I) Which kind of graphs have an "easy" counting procedure for the $F$-vector?

(II) Is there a natural classification for those graphs?

Question (I) is a source of inspiration for network reliability analysis. Here, we classify all graphs were "the $F$-vector is completely known". They represent the "easiest" counting problems under study. Then, we introduce a "level of difficulty", which is a function $l$ that assigns each graph an integer from the set $\{-1, 0, 1, \ldots\}$. Gracefully, "easiest" graphs have level of difficulty $-1$ or $0$, while non-easiest graphs assume positive level of difficulty. The main results only use the first theorem of Graph Theory, to know, Handshaking [**7**].

**2. BACKGROUND.** Your intuition is correct: if we delete "too many edges", the graph will be disconnected. On the other hand, if we delete none or few edges, normally the graph will be connected. In this section, both "corners" are formalized. We just state the results but do not spend time with the proofs, since they are thoroughly covered in the related literature.

All spanning trees have $n-1$ edges. Since they are minimally connected, there is no hope to find a graph with less than $n-1$ edges and to be connected. This implies that $F_x = 0$ for all $x > m-n+1$. Gustav Kirchhoff found in 1847 a closed formula to count the number of spanning trees in a given graph [**8**]. Specifically, the number of spanning trees, or "tree number of", $\kappa(G)$, is any cofactor of the Laplacian matrix of $G$ [1]. So, $F_{m-n+1} = \kappa(G)$, and since determinants are easy to find, we are able to count $F_{m-n+1}$ as well!

On the other hand, let us denote by $c$ to the minimum number of edges we must remove in order to disconnect the graph, sometimes called edge-connectivity, or simply connectivity. Naturally, by its definition we get that $F_x = \binom{m}{x}$ for all $x < c$. The minimum number of links required to disconnect two fixed nodes has been solved by Ford and Fulkerson [**5**], in the context of flows in networks [2]. Curiously, the number of ways to delete such $c$ links is a hard problem (again, in the class #P-Complete), but it is not hard for our problem. Michael Ball and Scott Provan [**2**] found the number $n_c$ of ways to delete $c$ edges and disconnect $G$. Therefore, $F_c = \binom{m}{c} - n_c$.

We summarize the known results on the $F$-vector with the following items:

- $F_x = \binom{m}{x}$ for all $x < c$.
- $F_c = \binom{m}{c} - n_c$.
- $F_{m-n+1} = \kappa(G)$.
- $F_x = 0$ for all $x > m-n+1$.

Observe that $m-n+1$ is precisely the difference between the number of edges of the graph and the ones of a certain spanning tree. This is the number of independent cycles, or the "co-rank" of $G$ [3]. We will denote it by $c(G)$. Clearly, if we delete more edges than the co-rank, the resulting graph is disconnected, since it has less edges than a tree. Therefore, $c \leq c(G)+1$. The key question for our graph classification is the following: is the equality possible? In that case, which graphs achieve the equality?

[1] This result is sometimes known as "Matrix-Tree Theorem", and connects algebra with graphs in an elegant way. Moreover, the birth of a new area of mathematics began with this result, called "Algebraic Graph Theory"

[2] Indeed, the authors found an algorithm to find the maximum flow between a source and a sink in a capacitated network with natural capacities. This is a foundational result of Flow Theory in networks.

[3] In algebraic graph theory, it is the dimension of the kernel of the incidence matrix for the graph $G$. For a neat cover of this topic, we invite the reader to see the book [**3**]

## 3. CLASSIFICATION.

**Definition (Level of difficulty).** The level of difficulty of a graph $G$ is the difference between its co-rank and connectivity:

$$d(G) = c(G) - c. \tag{2}$$

By previous observations, we know that $d(G) \geq -1$.

**Definition (Easy graph).** A graph $G$ is easy if $d(G) \leq 0$.

Observe that in easy graphs, the $F$-vector is fully known. In fact, consider a graph $G$ such that $d(G) = -1$. Then, $m - n + 1 = c - 1$. But we know all entries $x$ such that $F_x \leq c$ or $F_x \geq m - n + 1$, so the $F$-vector is fully recovered. The same situation occurs when $d(G) = 0$.

In this section, we will fully characterize connected graphs with $d(G) \in \{-1, 0\}$. Let us start with the easiest ones.

**Theorem 1.** *The level of difficulty is* $-1$ *only in trees and cycles.*

*Proof.* We study three possible cases:

1) If $c(G) = 0$ we have acyclic connected graphs, or trees. They have connectivity $c = 1$, and thus level of difficulty $-1$.
2) If $c(G) = 1$ we must have connectivity $c = 2$. For sure, the graph has no bridge (otherwise, the bridge disconnects the graph, and $c = 1$). Therefore, the minimum degree is $\delta(G) = 2$. Since $c(G) = 1$, we know that $m - n + 1 = 1$, or alternatively that $m = n$. By Handshaking, we get that $2m = 2n$ equals the sum of node-degrees. This implies that all nodes have degree 2, and the connected graph must be a cycle.
3) If $c(G) = i \geq 2$ we must have connectivity $c = i + 1$, and the minimum degree is therefore $\delta(G) \geq i + 1$. Since $c(G) = i$, alternatively $m = n + i - 1$. By Handshaking, we get that $2m = 2n + 2(i - 1) = \sum_{i=1}^{n} deg(v_i) \geq n\delta(G) = (i + 1)n$. If we subtract $2n$ on both sides, we get that $2(i - 1) \geq (i - 1)n$. Finally, dividing by the factor $i - 1$ we lead to the conclusion that a graph with co-rank $c(G) = 2$ and level of difficulty $-1$ must have $n \leq 2$ nodes. But this is impossible, since the minimum degree must respect $n - 1 \geq \delta(G) \geq 3$ in this case.

∎

Observe that during the proof of Theorem 2, only Handshaking Theorem was used. As a consequence, cycles and trees are the most elementary graphs under this new concept of difficulty. Let us go one step further, and characterize all graphs with level of difficulty $d(G) = 0$. The reader can have a try before reading. Indeed, the proofs follows the same spirit of Theorem 2, and Handshaking is the main tool. Before, we provide a hint. In 1990, Clyde Monma et. al. offered to the scientific community a foundational result from topological network design. He proved that the minimum-cost 2-node-connected spanning networks must either be a Hamiltonian tour or contain a special graph as an induced subgraph [**9**]. This special graph has been called Monma graphs for the first time in [**4**], to give the corresponding credits. They consist of two nodes connected by independent paths. Monma graphs are sketched in Figure 1. The hint: Monma graphs have the same co-rank and connectivity!

It is worth to mention some other graphs before the statement. When all nodes are directly connected, we have a complete graph. Butterflies are two triangles with a

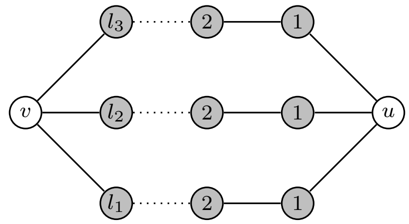


**Figure 1.** Monma's graph structure.

common vertex, and quasi-trees are graphs such that a tree is obtained once we remove a specific node (trees and cycles are omitted in this definition for convenience).

In order to illustrate the concept, let us find the level of difficulty for complete graphs with $n$ nodes, here denoted by $K_n$. The number of edges in $K_n$ is the number of ways to choose two nodes out of $n$. So:

$$d(K_n) = (n\frac{n-1}{2} - n + 1) - (n-1) = \frac{1}{2}(n-1)(n-4),$$

which is null if and only if $n = 1$ (trivial graph) or $n = 4$. The trivial graph is pathologic, in the sense that its connectivity is not defined. So, the only complete graph with null level of difficulty is $K_4$. The reader can have fun, finding the level of difficulty of the other graphs defined recently (Butterflies, Monma and Quasi-trees).

Yes... they have all level of difficulty $0$! We can say more.

**Theorem 2.** *The level of difficulty is* $0$ *only in Butterflies, Monma graphs, Quasi-trees and the Complete graph of order four.*

*Proof.* We should characterize the graphs such that the connectivity and co-rank are identical: $c(G) = c$. Since $c \geq 1$, we divide again the discussion into three possible cases:

1) If $c(G) = c = 1$, the graph must have at least a cycle and a bridge, leading in quasi-trees.
2) If $c(G) = c = 2$ we know that the minimum degree respects the inequality $\delta(G) \geq 2$. Since $c(G) = 2$ we get that $m = n + 1$. Now, Handshaking Theorem says that $2m = \sum_{i=1}^{n} deg(v_i) = 2n + 2$. Therefore, the graph must have either all but one or two nodes with degree two. This is a characterization Monma graphs and Butterflies, respectively.
3) If $c(G) = i \geq 3$, we have $\delta(G) \geq i$. Since $c(G) = i$, alternatively $m = n + i - 1$. By Handshaking, we get that $2m = 2n + 2(i-1) = \sum_{i=1}^{n} deg(v_i) \geq n\delta(G) = ni$. This means that $2(i-1) \geq n(i-2)$. Rewriting, we get that:

$$n \leq 2\frac{i-1}{i-2} = 2\left(1 - \frac{1}{i-2}\right) \leq 2\left(1 + \frac{1}{3-2}\right) = 4. \tag{3}$$

So, $n \leq 4$. But $n - 1 \geq \delta(G) \geq i \geq 3$. So, $n = 4$, and all inequalities are equalities. This is possible only in the complete graph with four nodes.

■

**Corollary 1.** Easy graphs are Cycles, Trees, Quasi-trees, Monma Graphs, Butterflies and the Complete graph with four nodes.

Corollary 1 is a full characterization of graphs whose $F$-vector is known a-priori. For completeness, we define easy reliability polynomials.

**Definition.** A polynomial is a reliability polynomial if it can be realized by a graph with Expression (1).

**Definition.** A reliability polynomial is easy if it comes from an easy graph.

**Corollary 2.** The complete list of all easy reliability polynomials is the following:

$$r_1(p) = p^m + mp^{m-1}(1-p)$$
$$r_2(p) = p^m$$
$$r_3(p) = [p^s + sp^{s-1}(1-p)]p^{m-s}$$
$$r_4(p) = p^m + mp^{m-1}(1-p) + (l_1l_2 + l_1l_3 + l_2l_3)p^{m-2}(1-p)^2$$
$$r_5(p) = p^6 + 6p^5(1-p) + 9p^4(1-p)^2$$
$$r_4(p) = p^6 + 6p^5(1-p) + 15p^4(1-p)^2 + 16p^3(1-p)^3$$

Reliability polynomials were presented in the respective order of the graphs from Corollary 1. A similar discussion can be performed to produce all graphs with a level of difficulty $d(G) = 1$. In order no to extend the discussion and invite further analysis for the reader, we prefer to draw all such graphs.

**4. CONCLUSIONS.** We played with counting problems in graphs to address a classical telecommunication problem from an abstract setting. As a corollary, we obtain a natural graph classification, to know, the level of difficulty, which is the difference between co-rank and connectivity. Graphs with a non-positive level of difficulty have the counting problem solved beforehand. They were fully characterized here. A full characterization of graphs with level of difficulty 1 is also feasible playing with Handshaking again, and we restricted to draw them. However, more sophisticated techniques should be developed to characterize graphs (and polynomials) with higher level of difficulty.

REFERENCES


1. Michael O. Ball. Computational complexity of network reliability analysis: An overview. *IEEE Transactions on Reliability*, 35(3):230 –239, aug. 1986.
2. Michael O. Ball and J. Scott Provan. The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM J. Computing*, 12:777–788, 1983.
3. N. Biggs. *Algebraic Graph Theory*. Cambridge Mathematical Library. Cambridge University Press, 1993.
4. Eduardo Canale, Pablo Monzón, and Franco Robledo. Global synchronization properties for different classes of underlying interconnection graphs for kuramoto coupled oscillators. In *Future Generation Information Technology*, volume 5899 of *Lecture Notes in Computer Science*, pages 104–111. Springer Berlin Heidelberg, 2009.
5. L. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.
6. E. N. Gilbert. Random graphs. *Annals of Mathematical Statistics*, 30:1141–1144, 1959.
7. F. Harary. *Graph Theory*. Addison-Wesley series in mathematics. Perseus Books, 1994.
8. G. Kirchoff. Über die auflösung der gleichungen, auf welche man bei der untersuchung der linearen verteilung galvanischer ströme geführt wird. *Ann. Phys. Chem.*, 72:497–508, 1847.
9. ClydeL. Monma, BethSpellman Munson, and WilliamR. Pulleyblank. Minimum-weight two-connected spanning networks. *Mathematical Programming*, 46(1-3):153–171, 1990.
10. Leslie G. Valiant. The complexity of computing the permanent. *Theor. Comput. Sci.*, 8:189–201, 1979.