A Graph Resilience Metric Based On Paths: Higher Order Analytics With GPU

Georgios Drakopoulos*§ , Xenophon Liapakis‡ , Giannis Tzimas† , and Phivos Mylonas*
*Department of Informatics, Ionian University
e-mail: {c16drak, fmylonas}@ionio.gr
‡Interamerican SA Greece
e-mail: liapakisx@interamerican.gr
†TEI of Western Greece, Antirrio Campus
e-mail: tzimas@teimes.gr
§Cloudminers Inc.

***Abstract*—Structural resilience is an inherent, paramount property of real world, massive, scale free graphs such as those typically encountered in brain networks, protein-to-protein interaction diagrams, logistics and supply chains, as well as social media among others. This means that in case a small fraction of edges or even vertices with their incident edges are deleted, then alternative, although possibly longer, paths can be found such that the overall graph connectivity remains intact. This durability, which is constantly exhibited in nature, can be attributed to three main reasons. First, almost by construction, scale free graphs have a relatively high density. Moreover, they have a short diameter or at least an effective diameter. Finally, scale free graphs are recursively built on communities. As a consequence, the effect of a few edge or even vertex deletions inside a community remains isolated there as a rule and the effects of deletion are thus negated. Ultimately these properties stem from the degree distribution. In this conference paper is proposed a new, generic, and scalable graph resilience metric which relies on the weighted sum of the number of paths crossing certain vertices of great communication and structural value. Finally, the CUDA implementation is discussed and compared to a serial one in mex. The metric performance is assessed in terms of total computational time and parallelism.**

***Index Terms*—Graph mining, large graph analytics, graph resilience, recursive community structure, triangles, graph diameter, degree distribution, higher order analytics, path analytics, GPU computing, CUDA, mex, bitmap, data structures**

1. Introduction

Graph resilience or structural integrity is vital in a broad spectrum of applications including but not limited to brain connectivity, social networks, geolocation services, digital multimedia content generation and delivery, argument graphs, logistics and supply chain management, computer and data communication networks, and chemical engineering. It consists of the ability of finding alternative, though generally more costly, paths when edges or vertices with their incident edges are deleted from the graph. In this work the cost of a path is defined as the number of its edges. Note, however, that in general this is not the only way to define path cost. For instance, for path cost in fuzzy graphs see [1] and [2].

Factors contributing to graph resilience in the case of scale free graphs are the high density, the short diameter, and the recursive community structure. The first factor ensures that there are enough hub or authority vertices, namely vertices with high out- or in-degrees respectively. Thus, there are many vertices connecting various graph segments. Moreover, the small diameter ensures that relatively low cost alternative paths are availble. Finally, the nested community structure implies that, besides certain critical edges connecting remote and maybe isolated communities, the loss of an edge or even of a vertex and its adjacent edges will likely result in no serious loss of structural integrity. Recall that a scale free graph is described by a vertex growth function $g(\cdot)$ with the property that for any positive scaling β_0

$$g(\beta_0 s) \;=\; h(\beta_0)\, g(s) \tag{1}$$

where $h(\cdot)$ is a function independent of the argument s.

From a structural point of view, resilience as a graph problem bears some similarity with vertex centrality and graph partitioning. Concerning the former, the more resilient a graph is, then the fewer weak points is expected to have, namely at the very least few, if any, articulation points and simultaneously few low degree vertices. Regarding the latter, a graph with high structural integrity is expected to yield partitions of mixed quality because of the many alternative paths interconnecting the different graph segments.

The primary contribution of this conference paper is a new graph structural integrity metric based on the weighted sum of the number of paths crossing each vertex. Moreover, the CUDA implementation of this metric is algorithmically described and certain aspects of this implementation are discussed, especially concerning parallelism and memory footprint. The analysis will be based on the following three assumptions.

Assumption 1. The graph is undirected.

Assumption 2. The graph has a single component.

Assumption 3. Resilience has a structural meaning.

Although it is fairly easy to extend the proposed metric in a way that one of the first two assumptions or both are waived, the third assumption must be carefully taken into consideration. The reason is that, if functional aspects are to be added to the definition of resilience, then the role of these functions and which conditions can lead to the loss thereof must be throroughly examined and algorithmically expressed.

The remaining of this work is structured as follows. Section 2 provides an overview of edge fuzzy graphs. The proposed graph resilience metric is described in section 3 along with a general comment regarding the mandatory higher order nature of such metrics. Finally, section 4 outlines the datasets and the associated results, whereas section 5 recapitulates the findings and explores future research directions.

Capital boldface are reserved for matrices, small boldface signifies a vector, capital letters represent sets or metrics, and small letters indicate constants or functions. Table 1 contains the notation of this work.

TABLE 1. PAPER NOTATION.

Symbol	Meaning
$\triangleq$	Definition or equality by definition
$\deg(v_k)$	Degree of vertex v_k
$(v_1, \ldots, v_n)$	Path with vertices $v_1, \ldots, v_n$
$\|p\|$	Path length
ζ_0	Graph diameter
$\{s_1, \ldots, s_n\}$	Set containing elements $s_1, \ldots, s_n$
$\|S\|$	Cardinality of set S
$\mathbf{1}_n$	Vector $n \times 1$ of ones
$\|\mathbf{x}\|_2$	2-norm of vector $\mathbf{x}$
$\otimes$	Kronecker tensor product

2. Related Work

GPU computation has been long proposed as a means for scalable graph computations ranging from parallel random graph generation with predetermined up to an extent degree distributions [3] [4]. Parallel graph analytics include graph traversal [5], which can form the basis for parallel versions of BFS or DFS, and graph coloring [6], which can be used to efficiently estimate Jacobians in the iterative solution of sparse linear systems.

Community structure discovery and vertex centrality are both paramount concepts in graph mining as well as in appliciations such as social media analysis, brain network analysis, and protein-to-protein interaction networks. Although they are discrete concepts, they are related to graph resilience since they are factors directly influencing it. For instance, in social network analysis digital influence was shown to be related to smooth information diffusion over the network [7] [8].

Resilience is an important characteristic of massive scale free graphs allowing efficient information propagation even in the face of temporary absence of certain edges [9] [10]. This rerouting capability which is achieved with minimal effort ensures the continuation of communication throughout the network [11] [12]. Applications of resilience include communication networks [13], transportation networks [14], brain networks [15] [16] [17], and protein networks [18] [19] [20].

In addition to being an important problem of its own right, graph resilience is linked to a number of fundamental concepts of social network analysis. To begin with, resilience has been partly at least attributed to the modular and recursive community structure of large graphs which results in a combination of local communication redundancy between communities with long range communication through bridges connecting communities [21] [22]. Resilience is also related to account influence [23] [24] or authority [25] [26] in social media. Finally, reslience has connections to the minimum cut problem which in turn is the basis for graph partitioning algorithms [27] [28].

The latter are supported in the NetworkX library in the Python ecosystem, massive graph processing systems such as Google Pregel [29], deep learning frameworks such as GraphLab [30], and persistent data structures like the one proposed in [31]. Additionally, schemas for linked data such as JSON4LD can be extended to include fuzzy graphs. Finally, graph computations with GPU can be conducted with Theano [32] [33].

3. Graph Resilience

3.1. Degree Distribution

Degree distribution in scale free graphs is a crucial parameter for any kind of graph ranging from Erdös-Rényi graphs to hyperbolic geometric and scale-free ones. For the latter category, a number of distributions have been recently proposed [34]. The first is the binomial distribution which states for a graph $G = (V, E)$ the probability that the degree of a vertex v equals k is

$$\text{prob}\{\deg(v) = k; q_0\} = \binom{|V|}{k} q_0^k (1 - q_0)^{|V|-k} \quad (2)$$

where $0 \leq q_0 \leq 1$ is essentially a parameter controlling the graph density compared to that of the complete graph with the same number of vertices.

Since the empirical degree distributions are skewed and not symmetric, the Poisson distribution has been proposed as a model for the degree distribution for scale free graphs where now the probability that the degree of v equals k is

$$\text{prob}\{\deg(v) = k; \lambda_0\} = \frac{\lambda_0^k}{k!} e^{-\lambda_0} \quad (3)$$

In (3) λ_0 is a positive parameter controlling the shape of the distribution through the height and the width of its curve. Thus, it also defines the decay rate or the tail height of the distribution.

However, the Poisson distribution has two main drawbacks concerning the modelling of scale free graphs. First,

it models two event classes, namely the common and the uncommon ones[1]. Second, it has a relatively thin tail for most values of λ_0. On the contrary, empirical studies have shown there are three degree classes and none of them is uncommon as defined by the Poisson distribution. Moreover, the actual degree distribution has a heavy tail. Therefore, the power law distribution has been proposed where

$$\text{prob}\left\{\deg\left(v\right) = k; \gamma_0\right\} \;=\; \alpha_0 k^{-\gamma_0} \tag{4}$$

In (4) α_0 is a normalizing constant such that

$$\begin{aligned}
\alpha_0 \;&=\; \sum_{k=1}^{|V|-1} \frac{1}{k^{\gamma_0}} \\
&\approx\; \int_{x_1}^{x_2} x^{-\gamma_0} dx + \xi_0 \\
&=\; \begin{cases} \ln\left(\frac{x_2}{x_1}\right) + \xi_0', & \gamma_0 \;=\; 1 \\ \\ \frac{x_2^{-\gamma_0+1} - x_1^{-\gamma_0+1}}{-\gamma_0+1} + \xi_0', & \gamma_0 \;\neq\; 1 \end{cases}
\end{aligned} \tag{5}$$

where x_1 and x_2 are integration bounds strictly between 1 and $|V|-1$ selected by appropriate approximation criteria such as large set cardinality estimation [35], whereas ξ_0 and ξ_0' are correction factors. The exponent γ_0 characterizes an entire class of graphs and usually but not necessarily lies between 2 and 3. The first value implies that the mean value of the degree distribution is bounded, while the second value means that the variance is also bounded.

3.2. Triangles

Conceptually it is fairly easy to understand the recursive nature of communities. For instance a cultural group in social media can be further subvided to literature, theater, dance, and motion picture subgroups which in turn can be further split recursively until some small and quite compact communities are reached. However, in practice it is not trivial to algorithmically distinguish two communities and so far a number of criteria have been proposed.

On the contrary, closed triangles, namely the elementary communities upon larger ones are recursively built, are tracked easily. Furthermore, their number in a graph can be computed using the graph adjacency matrix. Recall that the latter, denoted by $\mathbf{A}$, is defined elementwise as

$$\mathbf{A}[i,j] \triangleq \begin{cases} 1, & (v_i, v_j) \in E \\ 0, & (v_i, v_j) \notin E \end{cases} \quad \in \{0,1\}^{|V|\times|V|} \tag{6}$$

Then, the number of triangles Δ is

$$\Delta \;=\; \frac{1}{3}\,\text{tr}\left(\mathbf{A}^3\right) \;\triangleq\; \frac{1}{3}\sum_{i=1}^{|V|} \mathbf{A}^3[i,i] \tag{7}$$

1. It is worth mentioning that Poisson used this distribution to study deaths caused from horse kicks in the Prussian cavalry. Such deaths are typically considered rare. It was used as a device of irony in Márquez's *One hundred years of solitude*.

Equation (7) holds as each entry of $\mathbf{A}^\ell[i,j]$ by construction holds the number of paths of length ℓ from v_i to v_j. And a triangle is a closed path of length 3 from a vertex to itself. Thus, the number of triangles each vertex participates to is readily available. The only subtle point is that the trace counts each triangle three times, one for each of its vertices. Therefore, dividing each trace by three yields the total number of triangles in the given graph.

As a sidenote, there is an manifestation of the connection between structural graph resilience and higher order analytics. The former cannot be expressed in terms of first or second order analytics as the combinatorial properties of single vertices or single edges respectively are insufficient to describe the global property of structural integrity. On the other hand, third order analytics such as triangles can and probably higher order patterns such as stars, lines, circles, and cliques of variable sizes can be used to gain insight to the inner workings of large graphs. This is attributed primarily to the fact that a graph is by construction a distributed combinatorial object. In a scale free graph small world effects like a small diameter ζ_0 of $\text{O}\left(\log\log|V|\right)$ or a large number of high degree vertices reinforce the interplay between graph segments. Thus, local properties tend to be similar to global ones, which implies that even third, fourth, or fifth analytics may be able to capture global graph properties. The ability to deduce global properties from local ones in scale free graphs relies heavily besides the nature of the analytics on the graph topology and, thus, on the degree distribution.

3.3. Resilience Metric

Since each concept necessary to understand the proposed metric intuitively has been explained, it is high time for the metric to be explained algorithmically.

First, the number of both open and closed paths

$$P \;=\; \left(v_{j_1}, v_{j_2}, \ldots, v_{j_p}\right) \tag{8}$$

of length $|P| = p$ are computed, where p ranges from 3 to p_0. For open paths, $v_{j_1} \neq v_{j_p}$, while for closed paths, namely cycles, $v_{j_1} = v_{j_p}$. The intermediate vertices are all distinct, implying that

$$v_{j_2} \;\neq\; v_{j_3} \;\neq\; \ldots \;\neq\; v_{j_{p-1}} \;\neq\; v_{j_p} \tag{9}$$

Note that since each path contains p vertrices, it also has $p-1$ distinct edges as no vertex repetition is allowed, save for the last place. The length upper limit p_0 is $\Theta\left(\zeta_0\right)$ where ζ_0 is the graph diameter. Although ζ_0 may not be available, for scale free graphs it is known to be $\text{O}\left(\log\log|V|\right)$, which is suffices for building the proposed metric as p_0 simply needs to be $\text{O}\left(\log\log|V|\right)$ too.

Also, define the following sets

$$\begin{aligned}
\Sigma_p &\triangleq \left\{P \;=\; \left(v_{j_1}, v_{j_2}, \ldots, v_{j_p}\right); |P| \;=\; p\right\} \\
\Sigma_p(s) &\triangleq \left\{P \in \Sigma_p \cap s \in P\right\}
\end{aligned} \tag{10}$$

The first set Σ_p consists of all paths of length p, whereas the second set $\Sigma_p(s)$ comprises of all paths of length p which also have s.

Second, the number of triangles Δ is computed as mentioned in equation (7). This is tantamount to computing paths of the form

$$P' \ = \ (v_{j_1}, v_{j_2}, v_{j_3}, v_{j_1}) \tag{11}$$

Define the number of triangles which contain vertex v_k as

$$\Delta(v_k) \triangleq \mathbf{A}^3[k,k] \tag{12}$$

For each vertex v_k the following weighted sum is formed

$$\sigma(v_k;\mu_0) \triangleq \mu_0 \frac{\sum\limits_{p=3}^{p_0} \frac{1}{p-2}|\Sigma_p(v_k)|}{\sum\limits_{p=3}^{p_0} \frac{1}{p-2}|\Sigma_p|} + (1-\mu_0)\frac{\Delta(v_k)}{\Delta} \tag{13}$$

The interpretation of (13) is that it consists of the convex combination of two factors. The first is the of the fraction of the number of paths which contain v_k to the total number of paths of length up to p_0. The second factor is the fraction of the number of triangles containing v_k to the total number of triangles. The parameter μ_0 lies inclusively between 0 and 1. Also, $\sigma(v_k;\mu_0)$ by construction lies in the same interval.

The final step in order to obtain the proposed metric $J(\cdot)$ is to take the ratio between the maximum and the minimum values in order to assess the gap between the most and the less structurally important vertex of the graph

$$J(\mu_0) \triangleq \begin{cases} \dfrac{\max\{\sigma(s;\mu_0)\}}{\min\{\sigma(s;\mu_0)\}}, & \min\{\sigma(s;\mu_0)\} \ \neq \ 0 \\ 0, & \min\{\sigma(s;\mu_0)\} \ = \ 0 \end{cases} \tag{14}$$

The inspiration for (14) was the condition number κ_2 of a matrix, where the ratio of the largest to the smallest singular value determines the numerical condition of that matrix. In a similar manner, $J(\mu_0)$ is the relative distace between the maximum communication value a vertex can have to the minimum one. This depends on the parameter μ_0 as well as on the topology of the given graph. Thus, (14) achieves its objective as it is roughly inversely proportional to the structural graph resilience. The reason is that a large gap indicates that there is a vertex whose communication value is low. By the power law properties, the number of vertices with low communication value is relatively high, and, thus, there is a non-negligible fraction of vertices which contribute little to the overall graph coherence.

Some comments regarding (13) are in order. Notice that the contribution of paths are weighted by a factor which is approximately inversely proportional to their length. This is in accordance to the principle behind a large class of vertex centrality metrics which rely on matrix series such as the resolvent series $r(\cdot)$ proposed in [36]

$$r(s) \triangleq (\mathbf{A} - s\mathbf{I})^{-1} \ = \ \sum_{k=0}^{+\infty} \frac{1}{s^k}\mathbf{A}^k \tag{15}$$

The resolvent series reveal important information about the spectral properties of $\mathbf{A}$ and in general of bounded operators in Hilbert spaces. Among others, the resolvent series can be used to solve the inhomogeneous Fredholm integral equation

$$u(x) \ = \ g(x) + \lambda \int_{\tau_1}^{\tau_2} K(x,\tau)u(\tau)d\tau \tag{16}$$

The rationale behind indirectly penalizing long paths through weights is that shorter paths typically contribute more to graph resilience. This follows from the small world phenomena [37], the short diameter [38], the high density [39], and the large number of triangles in scale free graphs [40].

Notice that the first factor of (13) is reminiscent of the Newman metric. However, the latter is the ratio of the shortest paths which contain a specific vertex to the total number of shortest paths in a graph. Although the first factor of (13) can be considered a very crude approximation of the Newman metric, its origins are different and stem from a rewiring principle. Additionally, that factor of (13) can be efficiently computed in CUDA.

Additionally, the equation (13) is not a regularized equation but an ordinary convex combination as stated earlier. Also, it does not rely on intricate quantities as other metrics do, such as for instance the Estrada index, a resilience metric h proposed in [41] which requires that the spectrum of $\mathbf{A}$ be known. Specifically, the Estrada index is defined as

$$h \triangleq \sum_{k=1}^{|V|} e^{\lambda_k} \tag{17}$$

where λ_k are the eigenvalues of the adjacency matrix.

4. Results

4.1. Comparison To Serial Implementation

In order to assess both the total execution time and the actual parallelism gain, the proposed metric has been implemented both in CUDA and mex. The latter is the standard way to link and run C code inside MATLAB in order to create programs which combine the rich MATLAB native library collection with custom code which makes the most out of the underlying hardware.

The computation of $|\Sigma_p|$ is straightforward, since all is needed is the sum of all entries of $\mathbf{A}^p$. Furthermore, this sum can be efficiently computed in a parallel environment. However, the computation of $|\Sigma_p(v_k)|$ is quite different. Although the straightforward approach of explicitly constructing each $\Sigma_p(v_k)$ induces a significant overhead, this is compensated by the availability of multiple processing units. Additionally, the following three simple conditions which describe the incremental construction of $\Sigma_p(v_k)$ help identify the paths of interest. Empirically, for most scale free graphs, as the path length p increases, the number of paths of length p initially increase, then they reach a critical point, and then they start decreasing. Thus, it is relatively easy for

a GPU to count these paths, as, besides the part where each processor is assigned a set of paths to check for a specific vertex, the task is completely parallelizable.

The GPU in our disposal was a top end NVIDIA Titan Xp PG611-C00 with 12GB of GDDRX5 memory. Since this GPU yields 3840 CUDA cores and 240 texture units, the Titan Xp offers massive parallelism.

In order to assess the correctness and the speed of the proposed graph resilience metric, three families of synthetic graphs were examined. These families were generated by the Kronecker model, which leads to synthetic scale free graphs whose properties are almost identical to that of real ones. This model handles directly algebraically a given adjacency matrix $\mathbf{A}$, termed a generator matrix, in order to construct a series of related, self-similar adjacency matrices through the sequence

$$\mathbf{A}_{k+1} \;=\; \mathbf{A}_k \otimes \mathbf{A}, \qquad k \;\geq\; 1 \tag{18}$$

where $\mathbf{A}_0 \;=\; \mathbf{A}$. For the purposes of this work, three generators were used, shown in figure 1. From left to right, the generators 1a, 1b, and 1c have an increasing density, namely low, middle, and high.

The density ρ_0, namely the ratio of the number of edges to vertices, is a basic graph feature which nonetheless plays a significant role as it can be thought of as a crude approximation to average vertex degree.

$$\rho_0 \triangleq \frac{|E|}{|V|} \;\approx\; \frac{1}{|V|} \sum_{s \in V} \deg(s) \tag{19}$$

Notice that (19) is not the only graph density metric. For instance, the log-density ρ_0' is defined as the logarithm of the number of edges to the logarithm of the number of vertices

$$\rho_0' \triangleq \frac{\ln |E|}{\ln |V|} \;=\; \frac{\ln(\rho_0 |V|)}{\ln |V|} \;=\; 1 + \frac{\ln \rho_0}{\ln |V|} \tag{20}$$

In the important special case when $|E| \;=\; |V|^c$, then (19) and (20) respectively reduce to

$$\begin{aligned} \rho_0 \;&=\; |V|^{c-1} \;=\; |E|^{1-\frac{1}{c}} \\ \rho_0' \;&=\; c \end{aligned}$$

In order to understand the nature of the three graph families which will serve as benchmarks, ρ_0 and ρ_0' are shown. Figure 2 shows the actual density values ρ_0 for the first seven Kronecker graphs for each of the three sequence. Although in every case ρ_0 scales up linearly with $|V|$, the growth rate varies considerably. Thus, the three graph families are expected to exhibit different resilience properties, since the wiring in sparser graphs will be more likely to be severed from edge removals.

Additionally, in figure 3 the actual logdensity values ρ_0' versus the logarithm (base 4) of $|V|$ for the same graphs are depicted. The similar lines indicate that the same graph growth mechanism is in effect, including the log-density pivot points. Thus, the graphs may be parameterized by density or other factors of the generator graphs, but the underlying mechanism is the same. Therefore, they are not too dissimilar, at least as far as scaling is concerned, to prevent any meaningful conclusions to be drawn.

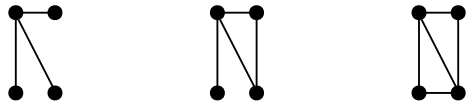
Given that information regarding the benchmark graphs, their resilience can be evaluated with the proposed metric and the results can be analyzed under that prism. Figure 4 depicts the normalized values of $J(\mu_0)$, meaning that the each resilience value in all three graph families has been divided by the smallest of them. This was done on purpose, as a common scale, defined by the smallest attained resilience value, would indicate similarities and differences not only among the same graph family but also among families. At the same plot the metric values for $\mu_0 = 0.5$ and $\mu_0 = 0.75$ are shown. Notice that in figure 4 the x-axis is the same as in figure 3. This was done on purpose to avoid the plot being overstretched. However, please keep in mind that x-axis is essentially logarithmically scaled.

Regarding the relative resilience values, the results are in general aligned to intuition as sparser graphs tend to be considered less resistant to edge or vertex removals than denser ones. However, this gap tends to become somewhat smaller when the graphs grow larger. This can be attributed to the small world phenomena that appear in larger free scale graphs. This interpretation is supported by the fact that for $\mu_0 = 0.5$, which gives to triangles equal importance with nearly all other paths, the gap is even smaller. On the other hand, for $\mu_0 = 0.75$, which considers vertex paths three times more significant than the triangles, the gap between the more and the less resilient graphs opens up about 1.2 and 1.5, roughly the half value of μ_0. Again, gaps tend to decrease with the number of vertices.

Now the question of scalability can be properly addressed. Figure 5 shows the relative speedup in total execution time, namely the execution time of the serial implementation divided by the time of the parallel one for the same number of vertices. Please notice that again the x-axis is logarithmically scaled. The adjacency matrices were small enough fit at the GPU memory and most of the code was written with that crucial fact in mind. As expected, the speedup is big for small and sparse graphs as there are fewer and shorter paths to count. On the contrary, the larger the graph grows or the denser it becomes, the gap between the two implementations closes. Nevertheless, the speedup was significant even in these cases. Notice that the value of μ_0 does not affect the computational time of these experiments.

Besides the advanced technology of the GPU and its memory capacity, an important speedup factor was that the vast majority of the task is easily subdivided to smaller and parallel and independent tasks. Thus, the only limiting factors are essentially the graph size, the number of threads, and any synchronization barriers. This is in accordance to Amdahl's law, where the overall speedup Γ_0 depends on the percentage π_0 of the task which can be actually parallelized as well as on the speedup of that part as follows

$$\Gamma_0 \triangleq \frac{1}{(1-\pi_0) + \dfrac{\pi_0}{\Gamma}} \tag{21}$$



(a) low (b) mid (c) high

Figure 1. The generator graphs.

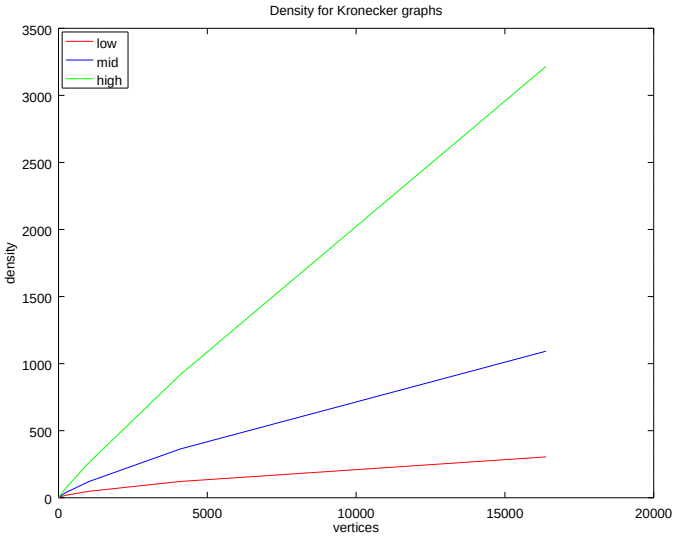


Figure 2. Density values ρ_0.

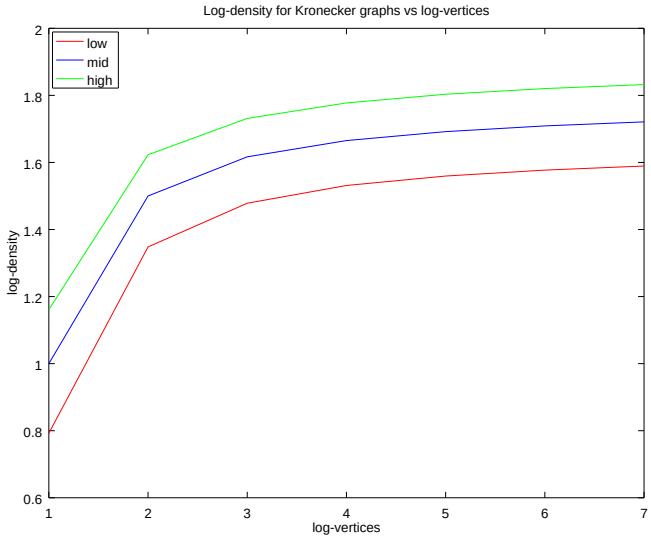


Figure 3. Log-density values ρ'_0.

4.2. Comparison To The Estrada Index

The only question which remains to be addressed is how well does $J(\cdot)$ performs compared to an established structural graph resilience metric. Although the findings of the previous subsections are reasonable, a comparison to another metric can shed some light on the way the proposed metric works. In this case the Estrada index, presented at (17) has been selected both because of its intuitive meaning and of the empirically verified correct results which yields. In fact, in this subsection the approximation of equation (22) to the Estrada index will be also used in order to assess the performance of the proposed metric.

Although the computation of the exponential function may be prone to numerical errors depending on the argument and the implementation, it is assumed here that it is executed perfectly and, thus, no such errors are present. Keeping the g_0 largest of the (signed) eigenvalues yields the approximation

$$h' = \sum_{k=1}^{g_0} e^{\lambda'_k} \tag{22}$$

Here it is assumed that $\lambda_1 \geq .. \geq \lambda_{g_0} > 0$.

The scheme for computing h' is shown in algorithm 1 which relies on the power method. Although the largest eigenvalue of $\mathbf{A}$ is known to be positive from the Perron-Frobenius theorem, this cannot be said for the remaining eigenvalues. In fact, large scale free graphs have been in many cases empirically shown to have negative eigenvalues or even a segment of their spectrum alternating around zero. Thus, although the power method does find the largest eigenvalue, it does not guarantee to yield a positive one. Thus, it is expected that in some cases more than g_0 repetitions of it will be used. Nonetheless, the rank one matrix corresponding to a negative eigenvalue must be subtracted from $\mathbf{A}$. The selection of the initial $\mathbf{s}$ guarantees that there will be components from every available direction.

The relative distance between the values of $J(\cdot)$ and h' when the latter serves as the reference value is defined as

$$d_0(|V|;\mu_0,g_0) \triangleq \left| \frac{h' - J(\mu_0)}{h'} \right| \tag{23}$$

Figure 6 shows the relative distance between the proposed resilience metric and the Estrada index for the three graph families. The Estrada index was computed for $g_0 = \lceil 0.1|V| \rceil$, whereas $J(\cdot)$ was computed for $\mu_0 = 0.5$.

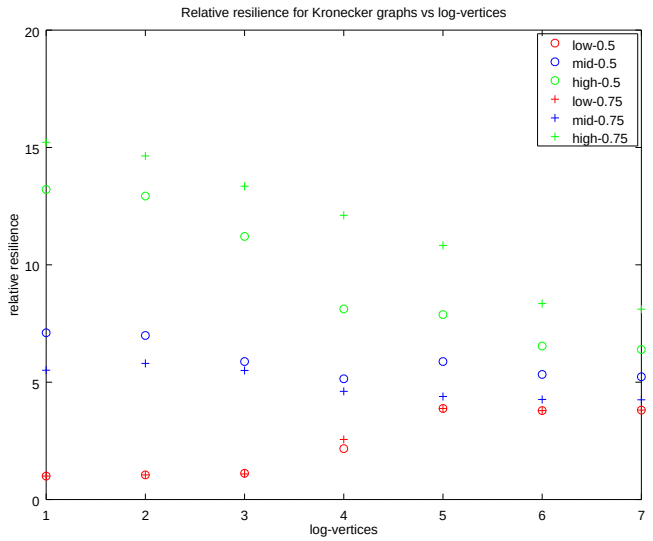


Figure 4. Graph resilience.

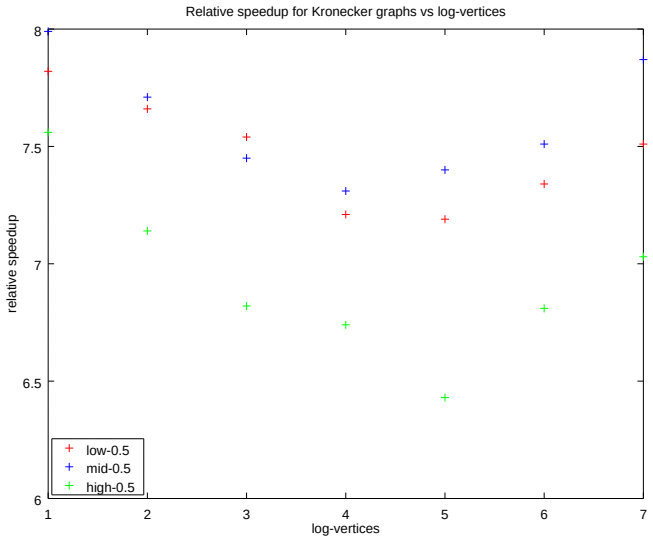


Figure 5. Metric scaling.

Although for small graphs there is a considerable distance, as graphs grow bigger this gap is reduced. This is attributed to the fact that the more paths are taken into consideration, more information regarding the structural coherence of a graph is revealed. Please note that also in this case the x-axis is logarithmically numbered.

5. Conclusions and Future Work

This conference paper presents a metric balancing paths and triangles, both fundamental elementary graph building blocks,for assessing the structural resilience of large graphs. Additionally, the speedup of a parallel implementation in CUDA for an NVIDIA Titan Xp is evaluated compared to that of a serial coding in mex, the standard way for linking C code to MATLAB. Besides the very satisfactory

Algorithm 1 Approximation (22) to Estrada index (17)

Require: Adjacency matrix $\mathbf{A}$; threshold η_0; eigenvalues g_0
Ensure: h' is computed as in (22)

repeat
  $\mathbf{s}^{[0]} \leftarrow \mathbf{1}_{|V|}/\sqrt{|V|}$
  repeat
    $\mathbf{s}^{[j+1]} \leftarrow \mathbf{A}\mathbf{s}^{[j]}$ **and** $\mathbf{s}^{[j]} \leftarrow \mathbf{s}^{[j+1]}/\left\|\mathbf{s}^{[j+1]}\right\|_2$
  until $\left\|\mathbf{s}^{[j+1]} - \mathbf{s}^{[j]}\right\|_2 \leq {}'\eta_0$
  $\lambda'_k \leftarrow \mathbf{s}^{[j+1]^T}\mathbf{A}\mathbf{s}^{[j+1]}$ **and** $\mathbf{A} \leftarrow \mathbf{A} - \lambda'_k\mathbf{s}^{[j+1]}\mathbf{s}^{[j+1]^T}$
  if $\lambda'_0 > 0$ **then**
    $h' \leftarrow h' + e^{\lambda'_k}$
  end if
until g_0 positive λ'_k have been found
return h'

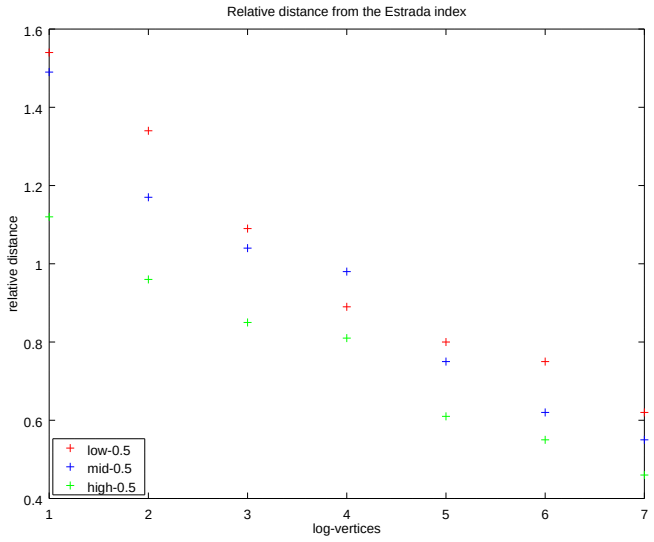


Figure 6. Relative distance to the Estrada index.

speedup, which can be attributed to the highly parallelizable nature of the proposed metric, the relative distance from an approximation of the Estrada index has been found to be within acceptable bounds. Three families of Kronecker graphs of various density were used as benchmarks in the experiments.

This work can be extended in a number of directions. An immediate task would be to explore the parameter space for values of the single parameter μ_0 which reveal important information about the graph. Simultaneously, as this work emphasizes on structural resilience, it makes sense to search for more elementary building blocks such as stars and squares to be included in the proposed metric. The relationshp netween graph connectivity patterns and the spectrum of the associated afjacency matrix must be also investigated. Needless to say that more experiments with larger graphs of finer granularity regarding their density are in order.

Acknowledgments

This conference paper is part of the project "Design and development of advanced software services for Interamerican SA".

Moreover, this work gratefully acknowledges the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for the purposes of this research.

Finally, this conference paper is part of Project 451, a long term research initiative whose primary objective is the development of novel, scalable, numerically stable, and interpretable tensor analytics.

References

[1] G. Drakopoulos, A. Kanavos, and A. Tsakalidis, "A Neo4j implementation of fuzzy random walkers," in *SETN*, May 2016.

[2] G. Drakopoulos, A. Kanavos, P. Mylonas, and S. Sioutas, "Defining and evaluating Twitter influence metrics: A higher order approach in Neo4j," *SNAM*, vol. 71, no. 1, 2017.

[3] S. C. Allmaier, S. Dalibor, and D. Kreische, "Parallel graph generation algorithms for shared and distributed memory machines," in *Advances in Parallel Computing*. Elsevier, 1998, vol. 12, pp. 581–588.

[4] S. Nobari, X. Lu, P. Karras, and S. Bressan, "Fast random graph generation," in *EDBT*. ACM, 2011, pp. 331–342.

[5] D. Merrill, M. Garland, and A. Grimshaw, "Scalable GPU graph traversal," *ACM SIGPLAN Notices*, vol. 47, no. 8, pp. 117–128, 2012.

[6] M. T. Jones and P. E. Plassmann, "A parallel graph coloring heuristic," *SIAM Journal on Scientific Computing*, vol. 14, no. 3, pp. 654–669, 1993.

[7] G. Drakopoulos, A. Kanavos, and A. Tsakalidis, "Evaluating Twitter influence ranking with system theory," in *WEBIST*, April 2016.

[8] A. Kanavos, G. Drakopoulos, and A. Tsakalidis, "Graph community discovery algorithms in Neo4j with a regularization-based evaluation metric," in *WEBIST*, April 2017.

[9] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, pp. 75–174, 2010.

[10] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over time: densification laws, shrinking diameters and possible explanations," in *KDD*. ACM, 2005.

[11] D. Loguinov, J. Casas, and X. Wang, "Graph-theoretic analysis of structured peer-to-peer systems: Routing distances and fault resilience," *IEEE/ACM transactions on Networking*, vol. 13, no. 5, pp. 1107–1120, 2005.

[12] B. Sudakov and V. H. Vu, "Local resilience of graphs," *Random Structures and Algorithms*, vol. 33, no. 4, pp. 409–433, 2008.

[13] J. P. Sterbenz, D. Hutchison, E. K. Çetinkaya, A. Jabbar, J. P. Rohrer, M. Schöller, and P. Smith, "Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines," *Computer Networks*, vol. 54, no. 8, pp. 1245–1265, 2010.

[14] W. H. Ip and D. Wang, "Resilience and friability of transportation networks: Evaluation, analysis and optimization," *Systems Journal*, vol. 5, no. 2, pp. 189–198, 2011.

[15] C. J. Stam and J. C. Reijneveld, "Graph theoretical analysis of complex networks in the brain," *Nonlinear biomedical physics*, vol. 1, no. 1, p. 3, 2007.

[16] A. V. Mantzaris, D. S. Bassett, N. F. Wymbs, E. Estrada, M. A. Porter, P. J. Mucha, S. T. Grafton, and D. J. Higham, "Dynamic network centrality summarizes learning in the human brain," *Journal of Complex Networks*, vol. 1, no. 1, pp. 83–92, 2013.

[17] E. Bullmore and O. Sporns, "Complex brain networks: Graph theoretical analysis of structural and functional systems," *Nature Reviews Neuroscience*, vol. 10, no. 3, p. 186, 2009.

[18] H. Jeong, S. P. Mason, A.-L. Barabási, and Z. N. Oltvai, "Lethality and centrality in protein networks," *Nature*, vol. 411, no. 6833, pp. 41–42, 2001.

[19] S. Maslov and K. Sneppen, "Specificity and stability in topology of protein networks," *Science*, vol. 296, no. 5569, pp. 910–913, 2002.

[20] M. E. Newman, "Mixing patterns in networks," *Physical Review E*, vol. 67, no. 2, p. 026126, 2003.

[21] J. Scott, *Social Network Analysis: A Handbook*. SAGE Publications Ltd., 2000.

[22] M. Newman, *Networks: An Introduction*. Oxford University Press, 2010.

[23] P. J. Carrington, J. Scott, and S. Wasserman, *Models and Methods in Social Network Analysis*. Cambridge University Press, 2005.

[24] J. Weng, E.-P. Lim, J. Lim, and Q. H. Jiang, "TwitterRank: Finding topic-sensitive influential twitterers," in *WSDM*, 2010, pp. 261–270.

[25] P. Jurczyk and E. Agichtein, "Discovering authorities in question answer communities by using link analysis," in *CIKM*, 2007, pp. 919–922.

[26] A. Langville and C. Meyer, *Google's PageRank and Beyond: The Science of Search Engine Rankings*. Princeton University Press, 2006.

[27] B. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *The Bell System Technical Journal*, vol. 49, no. 1, pp. 291–307, 1970.

[28] J. Shi and J. Malik, "Normalized cuts and image segmentation," *TPAMI*, vol. 22, no. 8, pp. 888–905, 2000.

[29] G. Malewicz, M. H. Austern, A. J. Bik, J. C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski, "Pregel: A system for large-scale graph processing," in *ICDM*, 2010, pp. 135–146.

[30] Y. Low, J. Gonzalez, A. Kyrola, D. Bickson, C. Guestrin, and J. M. Hellerstein, "GraphLab: A new parallel framework for machine learning," in *UAI*, July 2010.

[31] S. Kontopoulos and G. Drakopoulos, "A space efficient scheme for graph representation," in *ICTAI*. IEEE, 2014.

[32] F. Bastien *et al.*, "Theano: New features and speed improvements," *arXiv:1211.5590*, 2012.

[33] R. Al-Rfou *et al.*, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv:1605.02688*, vol. 472, p. 473, 2016.

[34] G. Drakopoulos, "Biological random graph models," April 2014, doi: 10.13140/RG.2.2.10774.45125.

[35] G. Drakopoulos, S. Kontopoulos, and C. Makris, "Eventually consistent cardinality estimation with applications in biodata mining," in *SAC*. ACM, April 2016.

[36] E. Estrada and D. J. Higham, "Network properties revealed through matrix functions," *SIAM Rev.*, vol. 52, no. 4, pp. 696–714, November 2010.

[37] P. Bonacich and P. Lloyd, "Eigenvector-like measures of centrality for asymmetric relations," *Social Networks*, vol. 23, no. 3, pp. 191–201, 2001.

[38] P. Bonacich, "Power and centrality: A family of measures," *American Journal of Sociology*, vol. 92, no. 5, pp. 1170–1182, 1987.

[39] S. P. Borgatti and M. G. Everett, "A graph-theoretic perspective on centrality," *Social Networks*, vol. 28, no. 4, pp. 466–484, 2006.

[40] J. Kepner and J. Gilbert, *Graph algorithms in the language of linear algebra*. SIAM, 2011.

[41] M. Benzi and P. Boito, "Quadrature rule-based bounds for functions of adjacency matrices," *LAA*, vol. 433, no. 3, pp. 637–652, 2010.