AFRL-RI-RS-TM-2018-001

EFFECTIVENESS OF VERTEX NOMINATION VIA SEEDED GRAPH MATCHING TO FIND BIJECTIONS BETWEEN SIMILAR NETWORKS

FEBRUARY 2018

TECHNICAL MEMORANDUM

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the 88th ABW, Wright-Patterson AFB Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TM-2018-001 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
ADNAN BUBALO
Work Unit Manager

/ S /
JON S. JONES
Technical Advisor, Information Intelligence Systems and Analysis Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
FEB 2018	TECHNICAL MEMORANDUM	SEP 2016 – SEP 2017

4. TITLE AND SUBTITLE

EFFECTIVENESS OF VERTEX NOMINATION VIA SEEDED GRAPH MATCHING TO FIND BIJECTIONS BETWEEN SIMILAR NETWORKS

5a. CONTRACT NUMBER
IN-HOUSE

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Tyler Witter

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
P1QW

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory/RIEA
525 Brooks Road
Rome NY 13441-4505

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RIEA
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RI

11. SPONSOR/MONITOR'S REPORT NUMBER
AFRL-RI-RS-TM-2018-001

12. DISTRIBUTION AVAILABILITY STATEMENT

Approved for Public Release; Distribution Unlimited. PA# 88ABW-2017-5146
Date Cleared: 20 Oct 2017

13. SUPPLEMENTARY NOTES
This Technical Memorandum represents independent government personnel evaluation of a tool developed under a DARPA sponsored Contract.

14. ABSTRACT

The purpose of this effort was to perform independent verification and validation of Vertex Nomination via Seeded Graph Matching, a graph analytic prototype developed under DARPA's XDATA program. The software was evaluated for its efficiency in applying principles of seeded graph matching to locating bijections among similar graphs. Contractor results were reproduced to verify integrity and an experiment was conducted to evaluate the performance of the software in a simplified environment where results were scored for various test cases. Issues and pitfalls in the underlying algorithms were pointed out along with areas of improvement for the current prototype.

15. SUBJECT TERMS

Vertex Nomination via Seeded Graph Matching (VN via SGM), Seeded Graph Matching (SGM), Vertex of Interest (VOI)

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			**ADNAN BUBALO**
U	U	U	UU	20	19b. TELEPHONE NUMBER (*Include area code)*

TABLE OF CONTENTS

Section **Page**

LIST OF FIGURES

Figure **Page**

1.0 SUMMARY

The development of graph matching heuristics is a venerable and active field for both commercial and military applications. Some of these include but are not limited to: two-dimensional and three-dimensional image analysis, document processing, biometric identification, image databases, video analysis, and biological and biomedical applications. [1] Graph matching has a specific military appeal as it can be utilized for building situational awareness, revealing plausible locations of enemy troops, supplies, fortifications, etc. With the advancements of modern day technologies, geospatial, temporal, and other forms of locational data are not only becoming more accessible, but the rate of influx is overwhelming analysts' abilities to analyze and make use of it. This heightens the demand for more optimal methods of data analysis, capable of pinpointing only relevant information for given task and providing faster processing rates while maintaining both efficiency and accuracy.

Vertex Nomination via Seeded Graph Matching (VN via SGM) is a newly developed graph matching algorithm prototyped in the R programming language and authored by developers at John Hopkins University under DARPA's XDATA program. [1] Modifying an existing graph matching algorithm, this software applies principles of seeded graph matching to locate bijections among similar graphs beginning at predefined seed nodes. The output is a probability matrix listing vertices nominated for their probabilities of being the correct match to some vertex of interest (VOI) within some target graph. Tailored towards large datasets, this algorithm attempts to reduce both problem size and processing time while maintaining performance and accuracy, by focusing on only subsets of the data relevant for a given task. The goal of this project was to evaluate the overall accuracy and efficiency of the VN via SGM algorithm by testing its performance in a simplified environment and scoring the results for various test cases. Hypothesis testing was then performed, analyzing the possibility that the algorithm may have alternate use cases in addition to locating a match to some VOI. Specifically, this vertex nomination process may provide analysts with important information about new datasets by exploiting its capacity for locating nodes with a degree of connectivity similar to the VOI. This provides a useful list of candidates an analyst can quickly examine to discover fruitful comparisons and noteworthy relationships within a graph. This could translate into a more rapid discovery of relationships among individuals, places, events, etc., providing a vehicle for improving the investigation and understanding of new data. To test this hypothesis, two vertices among our dataset were used as test cases for the VN via SGM procedure, and their output lists of vertex nominations analyzed for similar connectivity and relevancy to the VOI. All experiment results are then summarized and an overall evaluation is given to describe the proficiency of the current VN via SGM prototype.

2.0 INTRODUCTION

Given two graphs with the same total number of vertices, the goal of the most basic graph matching problem is to find bijections among similar graphs which minimizes the total number of adjacency disagreements. In this scenario, a bijection represents a one-to-one correspondence between all nodes in a graph, and an adjacency disagreement denotes the absence of an edge or relationship that should exist between two nodes. Central to the VN via SGM algorithm, is a variation of the graph matching problem known as the seeded graph matching problem (SGMP). Though the two are similar, the SGMP proceeds with one additional constraint – the bijection

function must first assign select vertices between the two vertex sets known as seeds, for which there is a known preexisting match. [1] To optimize problem size and processing time, the algorithm intelligently focuses on subgraphs surrounding the VOI and ignores the rest of the graph which is irrelevant for a given task. In this fashion, this algorithm supersedes other basic graph matching algorithms as it locates the same results with much improved efficiency. Using a modified preexisting seeded graph matching algorithm discussed in the next section, this current VN via SGM prototype aims to provide users with a tool for quickly locating bijections among similar graphs and nominating a list of possible matches to a given VOI within a target graph.

3.0 BACKGROUND

Suppose G_1 and G_2 are two graphs with vertex sets V_1 and V_2 such that they contain the same number of vertices. For any bijective function $\phi : V_1 \rightarrow V_2$, we can define the number of adjacency disagreements under ϕ by the following:

$$d(\phi) = \left|\left\{(u,v) \in V_1 \times V_2 : \left[u \sim_{G_1} v \text{ and } \phi(u) \not\sim_{G_2} \phi(v)\right] \text{ or } \left[u \not\sim_{G_1} v \text{ and } \phi(u) \sim_{G_2} \phi(v)\right]\right\}\right|. \text{[1]}$$

The graph matching problem therefore, is to minimize $d(\phi)$ over all possible bijective functions $\phi : V_1 \rightarrow V_2$. Since the simpler problem of deciding whether there exists a bijective function for establishing bijections between G_1 and G_2 is infamously of unknown complexity, minimizing adjacency disagreements over all possible bijective functions is undoubtedly NP-hard. More importantly, graph matching is notorious for having no efficient algorithms to date, and it is expected that none exist. [1]

Now consider the subsets $W_1 \subset V_1, W_2 \subset V_2$ where $|W_1| = |W_2|$ and we are given a fixed bijection $\varphi: W_1 \rightarrow W_2$. The SGMP, a variation of the graph matching problem, aims towards minimizing $d(\phi)$ over all bijections $\phi: V_1 \rightarrow V_2$ which are extensions of φ. In other words, ϕ must agree with φ on W_1, the elements of W_1 being the seeds of the graph. See Vogelstein et al. [2] for more details. Central to the graph matching process for this VN via SGM prototype, the authors modified the approximate graph matching algorithm of Vogelstein et al., [2] named Fast Approximate Quadratic Assignment Problem (FAQ), to be used for approximate seeded graph matching. In summary, they modified it to solve the relaxed SGMP using the Frank-Wolfe Method, an iterative procedure that involves successively solving linearizations and formulating them into a series of optimization problems. This results in a well-known linear assignment problem which is efficiently solvable in $O(n^3)$ time using the Hungarian Algorithm. This process is then repeated R times, to generate an average probability matrix which is used as an approximate seeded graph matching solution. [1]

4.0 EXPERIMENTATION AND PROCEDURE

4.1 Algorithm Overview

As mentioned in the introduction, VN via SGM is an exercise of the SGMP highlighting all possible bijections of some VOI and locating the best fit match within some target graph. With large datasets in mind, alongside inevitably inefficient runtimes of current graph matching

algorithms, the authors looked to optimize their algorithm by narrowing the size of the SGMP. By ignoring the vertices not important to the task, they focused their graph matching efforts on a subgraph, more specifically, some h-hop neighborhood around the VOI. In turn, the correct match if one exists must therefore be within some k-hop neighborhood, where $k \geq h$, of the seed nodes in the target graph. Through these assumptions, instead of wasting resources processing entire graphs, the algorithm instead focuses on only the portions of the graph necessary for locating the correct match. This significantly increases runtime efficiency without sacrificing large amounts of accuracy.

Given two graphs $G = (V, E)$ and $G' = (V', E')$, we denote $x \in V$ to be the VOI in G, where $x' \in V'$ represents its corresponding match in G'. Let $S \in V$ and $S' \in V'$ be the known seeds between the two graphs, with one-to-one correspondence $S \leftrightarrow S'$ and size $|S| = |S'| = s$. We now define W and W' as the remaining shared vertices, and J and J' as the remaining unshared vertices in G and G' respectively, where we denote $|W| = |W'| = n$, $|J| = m$ and $|J'| = m'$. Thus, we have $|V| = 1 + s + n + m$ and $|V'| = 1 + s + n + m'$. VOI x in G, and the correspondence of $S \leftrightarrow S'$ are the only information known to us a priori. Unknown to us however, are the total number of vertices $n, m,$ and m' or which vertices in G are shared or unshared. Given our search for the match to x, we can only assume that x' actually exists within G'. If not, we will continue our search for the most likely candidate whether it be the correct match or not.

At this point, the VN via SGM procedure has all the information it needs to locate candidates for x'. Before applying the modified seeded graph matching algorithm discussed in section 3.0, the procedure now optimizes the size of the SGMP by inducing smaller subgraphs to focus on. Thus, let $U = \{x\} \cup S \cup W$ and $U' = \{x'\} \cup S' \cup W'$ such that $|U| = |U'| = 1 + s + n$. Now consider the induced subgraphs $H = \Omega(U)$ in G and $H' = \Omega(U')$ in G'. Since U is comprised of the set of shared vertices which have one-to-one correspondences, we can assume the two subgraphs share some formal relationship. This warrants the likelihood that only vertices within the subgraphs are candidates for being a match to x, and anything outside of this subset can be ignored for our purposes. Utilizing this methodology for large datasets greatly reduces the size of the SGMP necessary for locating accurate results, and minimizes large amounts of unnecessary overhead.

As highlighted previously, seed selection is randomized within this current VN via SGM prototype. In turn, let S be a sampling of arbitrary size s from U. Selecting the size of our desired h-hop neighborhood, where $h \in \mathbb{N}$ – let S_x contain the seeds within some h-hop neighborhood of our VOI x such that $S_x = S \cap N_h(x)$, where S is our seeds and $N_h(x)$ is the neighbors of x within some h-hop distance. Since S_x are shared vertices, let S'_x contain the corresponding seed nodes within G' such that $|S_x| = s_x = |S'_x|$. In the obvious case, $h \to \infty$ yields $N_h(x) = G$, in which our induced subgraphs are the same as the original graphs, and hence we've failed to reduce the size of the problem. Inversely, if h is too small and $s_x = 0$, then we have no seeds to proceed with and the rest of the procedure need not apply. Thus, we assume $s_x > 0$, otherwise h must be increased until there are enough neighbors within the given h-hop neighborhood to satisfy the constraint.

Once we have confirmed our seeds where $S_x = S'_x$, located within H and H' respectively, we now assume x' to be within some k-hop neighborhood of our seeds in G'. Thus, let $C'_x = N_k(S'_x)$ contain the candidates for x' for $k \geq h$. If $x' \notin C'_x$, then x' doesn't exist within k-hop neighborhood of S'_x and the nomination list will not contain the correct match, but rather a list of

vertices that are most similar. In turn, k must be increased until x', if exists, is contained within C'_x. Lastly, subgraphs $G_x = \Omega(N_k[S_x])$ and $G'_x = \Omega(N_k[S'_x])$ are constructed containing the aggregation of S_x, x, and C'_x within G_x, and S'_x, x, and C'_x within G'_x. The modified SGM algorithm is now iterated a total of R times on the newly constructed subgraphs, and the results averaged into a final output probability matrix. Provided our assumptions are valid for a given input, we have now achieved a vertex nomination problem on a much smaller seeded graph matching problem.

4.2 Implementation

For this experiment, all tests were conducted on an Intel Core i7-2820QM 2.30 GHz quad core computer with 8 GB of RAM running Windows 7. The SGM algorithm is iterated a total of 100 times to construct an accurate average probability matrix as output. The number of seeds and the sizes of our h-hop neighborhoods are altered for each trial, and the environment is modified to test how well the algorithm performs under various conditions. The results are then compared in terms of the accuracy of the output, and the runtime efficiency of the algorithm for specific input parameters.

4.2.1 Author's Example

An example model provided by the authors at John Hopkins University, features a simple example to display the capabilities of their software on a small randomly constructed dataset. Laying the groundwork for this experiment was the production of ρ-correlated random dot product graphs (RDPG's). The process was then conducted as follows:

1. RDGP's $G(V, E)$ and $G'(V', E')$ are constructed, where $|V| = |V'| = 30$.
2. 5 vertices were removed from G' to make $|V(G)| \geq |V'(G')|$ to differ the graphs.
3. VOI x and s were randomly selected from the shared vertices.
4. S_x is selected from $N_h(x)$ in G, which matches S'_x in G' where $s_x = |S_x| = |S'_x|$. If $s_x = 0$, then the algorithm quits and returns "impossible1" meaning SGM will not run.
5. $C'_x = N_k(S'_x)$ contains the candidates for the match x' to x, for $k \geq h$. If $x' \notin C'_x$, then the algorithm quits and returns "impossible2" meaning SGM will not run.
6. Instantiate $G_x = \Omega(N_k[S_x])$ and $G'_x = \Omega(N_k[S'_x])$.
7. Run SGM($G_x, G'_x, S \leftrightarrow S'$) which returns $P = |V_x| \times |V'_x|$.
8. Locate $x' = max_{v \in C'_x} P[x, v]$. [3]

Running this example from start to finish gives us the following graphs in Figures 1 and 2 constructed in R, which illustrate the results of the output probability matrix when x is set to be node 22.

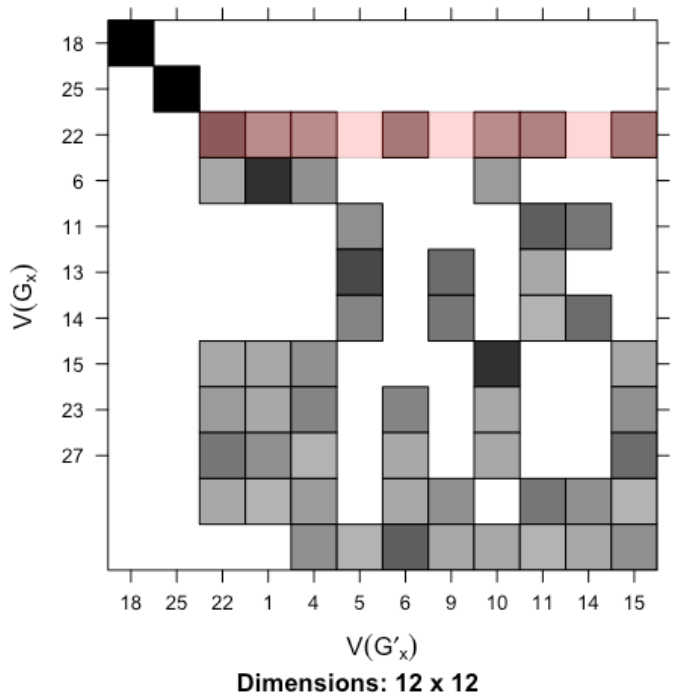

Figure 1: Submatrix of a probability matrix output from SGM

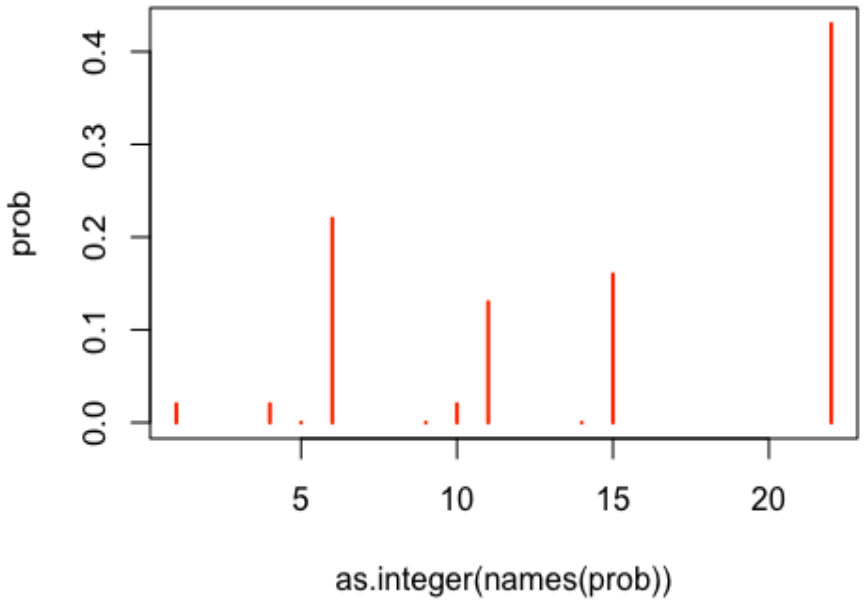

Figure 2: A bar plot of the matching probability of the candidates

Figures 1 and 2 above, contain two different ways of visualizing the nomination list given by the output probability matrix. Figure 1 represents a submatrix of the probability matrix, where rows 1 and 2 are the seeds, and row 3 is the VOI x. The shaded area in pink depicts the matching

probabilities of vertices in C'_x against x, where the darker the shade, the higher the probability that node is to being the best match. Figure 2 focuses on row 3 of Figure 1 in much more fidelity with a more familiar bar plot. In both cases, it is evident that node 22 has the highest probability of being a corresponding match to x while nodes 6, 15, and 11 are the next best candidates.

From this example, we can see that the VN via SGM algorithm was successful in finding a nomination list of candidates with the highest probabilities of being a match to x. This example however, was specific to the node 22 as the VOI and seed nodes 1, 8, 18, 25. Running this simulation multiple times, we come across many situations where there are no seed nodes within some h-hop distance of the VOI due to the random sampling of seeds. Increasing the size of h resolves this issue but at additional costs to the requisite processing times. Otherwise, if there are multiple seeds within some h-hop distance of the VOI, then the procedure runs successfully and returns an output probability matrix. Thus, this example lends credibility to this prototype as it successfully implements seeded graph matching on a smaller subset of the data and returns consistent results for various VOI's and given inputs.

4.2.2 Simple Example & Runtime Results

To best exercise the accuracy and efficiency of the VN via SGM algorithm, we simplified the environment and tested the algorithm's capacity for correctly identifying the match to any VOI when G and G' are identical. Tailoring the experiment towards big data, we chose the Visual Analytics Science & Technology (VAST) 2014 dataset as our test graph containing 1,480 nodes, 16,565 edges and average degree (number of edges incident to each vertex) of 22. Additionally, a small modification was applied to the current implementation to account for the case when randomly selecting seeds provides no seed nodes within a given h-hop neighborhood of the VOI. For our purposes, we will force the random seed selection from within the provided h-hop neighborhood of our VOI. In this manner, we have full control over the test environment without altering any assumptions or methodologies of the original procedure.

After loading the dataset into R and instantiating G and G', a similar methodology as the author's example was conducted to prepare this dataset for processing via the SGM algorithm. Since G and G' are identical, the set of shared vertices are all nodes contained in both graphs. This provides us the freedom of testing the accuracy and efficiency of this vertex nomination process for every node as the VOI, while knowing exactly what the correct match should be. Thus, we tasked the algorithm with locating the correct match to every possible VOI in G while varying initial quantities of seeds, and sizes of the induced subgraphs by altering the searchable h and k-hop neighborhoods. In short, we expected both the number of seeds, and the size of the h and k-hop neighborhoods to play major roles in its ability to consistently locate the correct match to the VOI in an efficient runtime. Results from each trial tell us how varying the initial conditions influences the accuracy and total required runtime of the procedure for specific inputs.

4.2.2.1 Trial with One Seed and One h-hop Neighborhood

Iterating through every node as the VOI, with only one initial seed provided us with a variety of results. To best optimize the performance of the algorithm, we kept the searchable h and k-hop neighborhood sizes to one. Since we know the correspondence of the two graphs, the match

to each of the VOI's is guaranteed to exist within a one hop neighborhood of our seeds. Iterating through all 1480 nodes as our VOI, the process successfully ran and produced a nomination list for every vertex that had at least one neighbor within a one hop distance. From the total 1480 iterations, 216 of these failed to find the correct match to the VOI. Repeating the test multiple times to ensure consistent results, the VN via SGM algorithm found an incorrect match to the VOI 20% of the time under these parameters. The average runtime required to calculate one nomination list was 6.3 seconds for an average induced subgraph size of 37 nodes. For induced subgraphs of size 10 or less, the runtime required was less than 0.3 seconds. Induced subgraphs of 100 nodes required about 10.5 seconds of runtime and highly connected VOI's with induced subgraphs of 200 nodes, required one minute and 40 seconds of processing time. Using Microsoft Excel to generate a best fit curve on the results, Figure 3 below details the cubic polynomial relationship between the size of the induced subgraphs and the processing time required for a given input size. This curve aligns perfectly with the expected complexity of the SGM algorithm discussed in section 3.0, validating the credibility of our results.

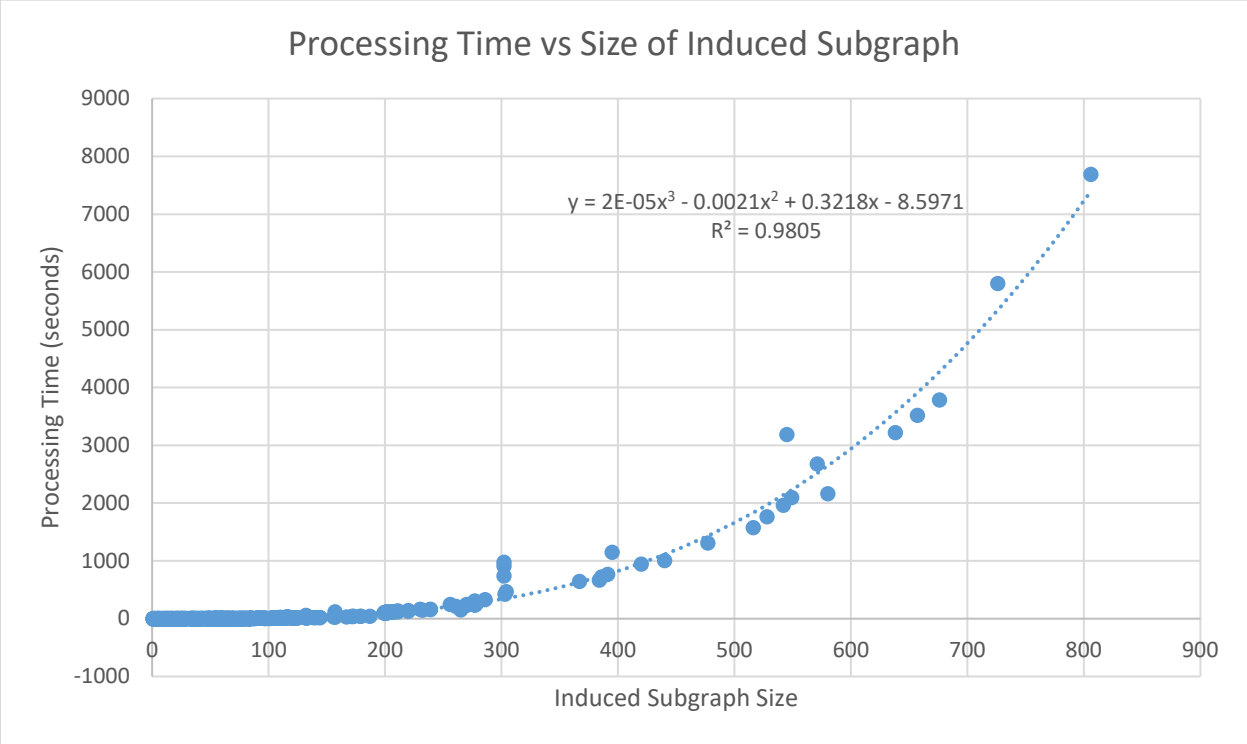

Figure 3: Scatter plot containing cubic polynomial relationship between processing time of the VN via SGM algorithm and size of the induced subgraphs

Using the curve in Figure 3, we can approximate how this algorithm scales for larger machines. For an induced subgraph size of 300 nodes, this algorithm required approximately 500 seconds of processing time. Using the curve, we can conclude that a machine twice as powerful could process approximately 420 nodes in the same period and a machine ten times as powerful could process around 720 nodes. Thus, we have a nonlinear relationship between performance capabilities and a machine's computational power. If an analyst was looking to improve the

accuracy of their results, it is necessary to either acquire more seeds or increase the size of the h and k-hop neighborhoods for which seeds can be selected. The sacrifice however, are larger induced subgraphs resulting in inefficient runtimes as expressed in Figure 3. Thus, there exists an optimal balance between the number of initial seeds and the size of the h and k-hop neighborhoods for achieving maximum accuracy.

4.2.2.2 Trial with Two Seeds and One h-hop Neighborhood

Initializing this process again with two initial seeds delivers results as we would expect comparative to one seed. From all 1,480 total iterations, the results denote a total of 869 successfully constructed vertex nomination lists, 48 of these indicating an incorrect match to the VOI (~6%), and 611 failures to run the algorithm. Using two seeds, the average size of the induced subgraphs increased to 59 nodes (from 37) and the average runtime required was now 11.27 seconds per calculation which is over double the average runtime required when using one seed. We notice however, a large increase in accuracy using two seeds in which only 6% of the iterations which provided outputs was unsuccessful in locating the correct match to the VOI. Hence we increased accuracy at the expense of more overhead among total calculations.

The nodes which failed to produce results were due to a lack of sufficient neighbors to the VOI to elect as seeds. Thus, if there is a lack of connectivity to our VOI, either the accuracy of the VN via SGM algorithm sharply decreases or it fails to run completely. Substantial connectivity on the other hand had a different effect on the algorithm. Among the nodes which resulted in an incorrect match to the VOI, the majority had a large 1-hop neighborhood. More importantly, if we vary the two seeds chosen among this neighborhood, the procedure locates a completely different match to the VOI. In one test case for example, varying our initial two seeds for one specific VOI, resulted in 6 different end solutions for the correct match to the VOI with a 99% probability that each one of these matches was correct. This is a usability concern to take note of, which argues for the importance of choosing the best possible seeds, and optimal number of seeds for a given VOI.

4.2.2.3 Trial with Five Seeds and One h-hop Neighborhood

Assuming we have a highly-connected graph, finding the match to a given VOI should be a much more effective and reliable process. Repeating the same experiment with five seeds warrants a successful operation for all nodes with h-hop neighborhood sizes of five and larger. Out of the 310 successful iterations of the algorithm, there were only four failures to find the correct match to the VOI (~1%). Although the accuracy has improved, the average runtime of these 310 processes was calculated at 19.61 seconds for an average induced subgraph size of 95 nodes which is a clear increase from the previous test cases using fewer seeds. Hence, the more seeds we allow for – the larger the SGM problem is, and the more consistently accurate the results are at the sacrifice of longer runtimes.

4.2.2.4 Trial with a Dynamic Number of Seeds and One h-hop Neighborhood

To test the total number of seeds roughly necessary for providing accurate results for a given VOI, we halved the total quantity of its neighbors within a one hop neighborhood. Taking

the ceiling of this number as our total number of seeds, we randomly selected seed values from the same set of neighbors and conducted the VN via SGM procedure comparing its output to the results from previous trials. After all 1,480 iterations, the correct match to the VOI was located 112 times giving us a failure rate of 7.5%. This was much improved from the trial in section 4.2.2.1 which featured only one seed by 12.5%. The average runtime however, increased from 6.3 seconds to 20.3 seconds and the average size of the subgraphs increased from 37 to 60 nodes. Among the iterations that failed to find the correct match, most the VOI's had limited edge connectivity. Thus, our results are hindered by the nodes with lesser degrees of edge connectivity and in this test case can only be compared with the output from section 4.2.2.1.

Running the same trial again but limiting the scope to VOI's with at least five neighbors within a one hop distance, we can now compare the results to the trial in section 4.2.2.3 and assess if using half the number of available neighbors as our total number of seeds is an effective threshold for generating improved results. In this experiment, the algorithm failed to find the correct match to the VOI only 3 out of 310 times – a notable 1% failure rate. This wasn't an improvement from the trial in section 4.2.2.3 which was also showed a 1% failure rate. The average runtime required was now 63.8 seconds compared to 19.61, and the average subgraph size was now 129 nodes versus the 95 nodes of the latter. Thus, using a quantity of seeds equal to half the number of available neighbors within a one hop distance of the VOI, is a successful but suboptimal method for generating improved results as it suffers in runtime efficiency. We have also narrowed the threshold of the optimal number of seeds necessary for providing accurate results and efficient runtimes to between two and five depending on the graphs and VOI's in focus.

4.2.3 Impact of Differing the Graphs

Now that we have determined some efficiencies and drawbacks of the VN via SGM procedure on a trivial environment, we tested a more practical situation to see how the algorithm performs when the graphs in comparison are not isomorphic. To simulate this using the same dataset, 150 nodes were deleted from the target graph and the same experimentation as detailed in section 4.2.2.2 was conducted using two seeds and a h-hop neighborhood of size one as input. From the total 1,480 iterations, 790 successfully returned an output nomination list, 548 terminated prior to completion, and 122 of the 790 failed to locate the correct match to the VOI. The 548 early terminations were once again, the result of 548 VOI's not having at least two immediate neighbors to elect as seeds. The average induced subgraph contained at least 48 nodes and required 17 seconds of runtime to process which is reasonable compared to previous results. In comparison to the two seed trial in section 4.2.2.2, the deletion of 150 nodes from the target graph caused some major overall issues for the VN via SGM procedure as we would expect, triggering an uptick of total incorrect matches from 6% to 15.4%. In summary, the less isomorphic the graphs are, the more difficulty this procedure has in locating the correct match to a VOI and generating an accurate probability matrix as output.

4.2.4 Increasing the Size of the h & k-hop Neighborhoods

So far, we have evaluated the effect that increasing the total number of seeds has on the VN via SGM procedure. If the degree of edge connectivity for our VOI is a quantity less than the desired number of seeds for some h and k-hop neighborhood, then failure rates increase due to the

resulting decrease in the number of seeds. To allow for more seeds, the searchable neighborhood about the VOI must be increased. Thus, we must increment the sizes of the h and k-hop neighborhoods until there are enough neighbors necessary for selecting the desired number of seeds from. Repeating the two seed trial from section 4.2.2.2 and increasing the size of the h and k-hop neighborhoods from one to two, resolved the issue causing the early terminations of the procedure for lesser connected VOI's. In return, this significantly amplified the size of the induced subgraph and thus the size of the SGM problem. The average size of the induced subgraphs was now a non-optimal 330 nodes requiring an average processing time of 19.3 minutes. Out of the 48 VOI's that were handled by the VN via SGM procedure in a total of 15 consecutive hours, 7 of these 48 iterations were unsuccessful in finding the correct match to the VOI (~14%). To strengthen the accuracy of the results, the number of requisite seeds must be increased. The result however, would entail an increased size of the SGMP and subsequent processing time to produce an output.

4.2.5 Comparison of Results

Table 1 below contains a collection of results from all experiments within section 4.2. Specifically, this table illustrates the relationship between the input values and the efficiency and accuracy of the VN via SGM algorithm. The smaller the graph size, number of seeds, and searchable h and k-hop neighborhoods around the VOI, the faster the processing time of the algorithm. The results however weren't as accurate for experiments using a larger number of seeds. As the number of seeds increased, the algorithm visibly featured better accuracy rates at the cost of longer runtimes. Dynamically allocating the number of initial seeds equal to half the total neighbors of the VOI, as seen in section 4.2.2.4 didn't improve the results compared to using two or five seeds, and increasing the h and k-hop neighborhoods to two decreased performance on both ends. Lastly, when we differed the two graphs by 150 nodes, the algorithm expectedly showed a large uptick in inaccuracy.

The last important notion to draw from these results, is that the algorithm was never 100% successful in locating the correct match to the VOI even in this simplified environment. Provided with an optimal number of seeds for a given VOI however, this algorithm will output probability matrices with less than 1% error within a reasonable length of time. Results from section 4.2.4 illustrate how long the process can take when the algorithm is unsuccessful in optimizing the scope of the problem and is forced to process large subgraphs or even the entire graph. This may occur more frequent than not when there a limited number of seeds for the algorithm to work with. Using at the results from section 4.2.4 as the worst possible case, this table demonstrates the success of the VN via SGM algorithm when analyzing large datasets. Provided some fine tuning of the algorithm in future prototypes, and usage of this algorithm on a more powerful machine, the results are subject to improve relative to this benchmark.

Table 1: Collection of experimentation results capturing the relationship between input to the algorithm and the performance measures

Experimentation Results For Various Input (Seeds, h & k)					
	Input		**Output**		
Trial	**Seeds**	**Size of h & k**	**Avg. Subgraph Size**	**Accuracy**	**Avg. Runtime**
4.2.2.1	1	1	37	80.00%	6.30 sec
4.2.2.2	2	1	59	94.00%	11.2 sec
4.2.2.3	5	1	95	99.00%	19.6 sec
4.2.2.4	Half the size of neighborhood	1	60	92.50%	20.3 sec
4.2.2.4	Half the size of neighborhood greater than 5	1	129	99.00%	63.8 sec
4.2.3	2	1	48	84.60%	17.0 sec
4.2.4	2	2	330	86.00%	19.3 min

4.3 Hypothesis Testing

To test our hypothesis, we needed to analyze whether the output of this VN via SGM algorithm can be utilized for more than just a list of possible matches to some VOI. More specifically, we seek to discover whether an analyst could use the output of this algorithm to quickly locate and extract important communities of nodes from new graphs. Communities in this sense, are subsets of nodes that are of significant importance to a network or graph, and may quickly point to areas where critical relationships may exist between actors in this network.

Selecting good test cases was imperative in this process. Examining the VAST 2014 Challenge dataset closely, we located a number of great candidate nodes to select as our VOI's. The selection criteria, being nodes that are highly connected within the dataset and may therefore be impactful among a network of individuals. Highlighting some background information for this data and the challenge itself, the data contains information about a network of individuals and the disappearances of two people from this network. There are two large organizations represented in this dataset which are responsible for clustering the nodes into two main groups. One organization is a company known as GASTech, and the other is an activist group called the Protectors of Kronos (POK). The POK is suspected in the disappearance, but conclusive evidence has yet to be gathered. Follow the reference to the VAST 2014 Challenge in the bibliography for more insight into the challenge and the dataset itself. [4] If our hypothesis is true, then this software might serve as a useful utility in conducting an initial investigation into these disappearances and discovering important communities of interest. The search criteria for selecting VOI's now extended to individuals who are strongly connected to either the GASTech Company, the POK, or uniquely associated to both. Thus, the specific nodes (individuals) chosen from the dataset for this experiment include Sten Sanjorge Jr., the CEO of the GASTech Company and Isia Vann, who is both a GASTech employee and POK member. In choosing vertices which meet the discussed

criteria, we expect the VN via SGM algorithm to locate other individuals who share similar degrees of connectivity among either or both organizations.

To launch the experiment, we once again duplicated the graphs in order to perform vertex nomination on this network of individuals. To achieve a wide breadth of results, the algorithm was implemented with h and k-hop neighborhoods of size two, creating large subgraphs surrounding the VOI as a result. Using five seeds, the algorithm found the correct individuals as matches to our VOI's with high confidence. Running the algorithm using Isia Vann as our VOI provides the following output displayed in Figure 4 below. The node with the highest probability as seen in the bar plot is in fact the correct match as it is the node representing Isia Vann in the target graph. The three other top ranked nodes in the output nomination list were examined in detail and were all found to have large connectivity comparable to that of Isia Vann. Digging a little deeper, it is apparent that these nodes all have a clear relationship with Isia Vann and play a major role in the rest of the events that occur in this dataset. These results were consistent with the trial we ran for Sten Sanjorge Jr. The algorithm found the correct match to our VOI with high confidence and the next best candidates shared a similar measure of connectivity to that of Sten Sanjorge Jr. Additionally, the next best candidates were all individuals or places that played a key role in other events unfolding within this dataset. From these results we argue that our hypothesis is true, validating VN via SGM's potential as a utility for analysts in quickly locating communities of similarly connected nodes among a graph and revealing important relationships and information about new networks of data.

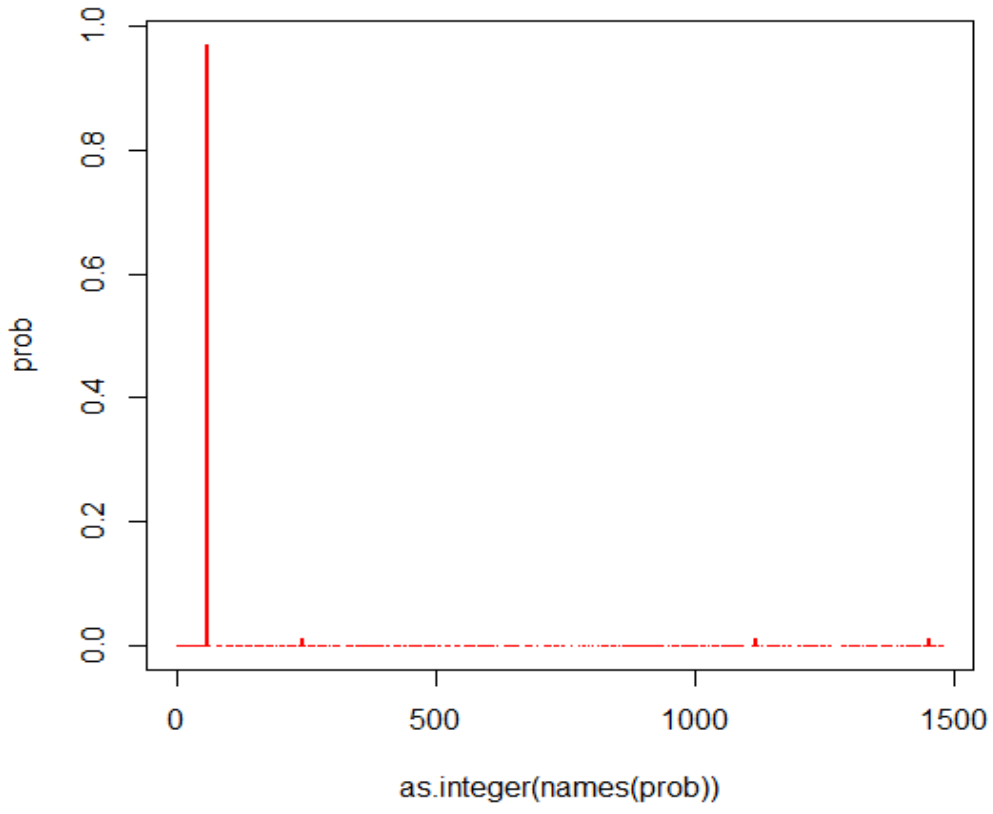

Figure 4: Bar Plot displaying candidate matches to Isia Vann as the VOI

5.0 DISCUSSIONS

The current VN via SGM prototype faces many obstacles that hinder its capacity for wide scale performance. Chief of them is the degree of similarity necessary between both graphs for achieving worthwhile results and subsequently identifying real world situations and test cases which meet this demand. Featuring graph matching logic based entirely around the connectivity of the graph and its individual parts, in general, the more disparate the graphs are the more inaccurate the results are. Secondly, the logic used to optimize performance via subgraphs is also greatly influenced by the connectivity of individual nodes in the graph as well as the seeds it is provided with. Choosing "good" seeds, or seeds that best capture the structure (similarity) of the graph plays a crucial role in the process. In the current implementation, the seeds are selected at random, thus providing no guarantee that they accurately capture the structure of the graph surrounding the VOI, or more importantly guarantee similar graph structure surrounding the VOI. The introduction of the Page-Rank algorithm is a possible resolution as it has been proven to be a great strategy for finding "good" and useful seeds. [5]

6.0 CONCLUSIONS

As mentioned earlier, graph matching is a computationally intensive process. The graph matching problem itself – minimizing the total number of edge disagreements over all bijective functions – is a NP-hard problem. Even the more elementary process of determining graph isomorphism between two graphs is notorious for unknown complexity. [2] Although polynomial algorithms exist for many graph matching problems (e.g., planar graphs, trees), it is suspected and generally understood that no efficient algorithms exist for performing graph matching. [1]

Provided with a favorable environment the VN via SGM algorithm is commendable in its ability to decrease the runtime required for performing graph matching. Given nearly isomorphic graphs and ample connectivity, this algorithm has proven itself in providing confident results given an optimal number of seeds and sufficient h and k-hop neighborhoods about the VOI. The smaller the size of the induced subgraph necessary for locating the correct match, the higher the runtime efficiency of the algorithm. The more optimal the seed selection is, the more accurate the solutions are. Thus, the major internal constraints are the initial number of seeds, the specific seed selections, and the minimal required sizes of the h and k-hop neighborhoods necessary for locating a match. When successful, the output probability matrix is very useful in locating which nodes among the candidate list are similar to the VOI. Using this fact to validate our hypothesis, this algorithm has also proven itself as a potential utility for analysts in quickly analyzing and extracting important relationships from new datasets.

A better method for testing the runtime performance of this software is to leverage a distributed computational environment. Although limited to the computational power of the computer used in this experiment, the results are expected to scale nicely to a more powerful machine or cluster. Varying the input parameters as seen throughout this experiment should have the same effects on the performance of the algorithm. Distributing the workload of similar experiments would significantly decrease required runtimes and allow for more rapid testing by researchers and analysts.

The current VN via SGM implementation is still in its prototyping stage and requires refining before it can be utilized by an end-user or Air Force Analyst. There are large number of situations for which the algorithm crashes and/or provides false positives based on the choices of seeds, and the connectivity of the graphs. Flaws were pointed out in this report describing its pitfalls in a simplified environment where it was provided with both similar and identical graphs. Locating real world situations containing graphs similar enough that this software can be utilized confidently, would be a difficult task.

7.0 FUTURE WORK

A worthwhile future experiment for testing the VN via SGM algorithm would include determining the optimal number of seeds and sizes of h and k-hop neighborhoods about the VOI necessary for locating consistent and accurate results in a minimal amount of time. Valuable information for analysts using this software would include the number of seeds and subsequent size of the SGMP necessary for achieving 100% accuracy when possible. Additionally, determining what degree of similarity between graphs is necessary for attaining 100% accuracy would be equally as valuable. Lastly, pinpointing some real-world applications and analyzing the algorithm's performance against real data would be highly beneficial for scoring the overall usability of the VN via SGM algorithm.

8.0 BIBLIOGRAPHY

[1] Fishkind, D., Adali, S., and Priebe, C., (2012). Seeded graph matching. <arXiv:1209.0367>.

[2] Vogelstein, J.T., Conroy, J.M., Podzarik, L.J., Kratzer, S.G., Harley, E.T., Fishkind, D.E., Vogelstein, R.J., and Priebe, C.E., (2012). Brain graph matching via fast approximate quadratic programming. <arxiv.org/pdf/1112.5507>.

[3] Priebe, C.E., Park, Y., Patsolic, H., Lyzinski, V., (2016). Vertex Nomination via Seeded Graph Matching. <http://www.cis.jhu.edu/~parky/XDATA/SGM/vn.html>.

[4] Unknown., (2015). VAST Challenge 2014 <http://www.vacommunity.org/VAST+Challenge+2014>.

[5] Moradi, F., Olovsson, T., Tsigas, P., (2014) A Local Seed Selection Algorithm for Overlapping Community Detection <http://www.cse.chalmers.se/~tsigas/papers/ASONAM14.pdf>.

9.0 LIST OF ACRONYMS

DARPA	Defense Advanced Research Projects Agency
GB	Gigabyte
GHz	Gigahertz
RAM	Random Access Memory
RDPG	Random Dot Product Graph
SGM	Seeded Graph Matching
SGMP	Seeded Graph Matching Problem
VAST	Visual Analytics Science & Technology
VN via SGM	Vertex Nomination via Seeded Graph Matching
VOI	Vertex of Interest