AFRL-RV-PS-
TR-2022-0062

AFRL-RV-PS-
TR-2022-0062

A MULTISTAGE STOCHASTIC OPTIMIZATION APPROACH FOR IDENTIFYING STABLE AND INFLUENTIAL CLUSTERS IN RANDOMLY CHANGING NETWORKS

Maciej Rysz

Miami University
501 E. High Street
Oxford, OH 45056

18 March 2022

AIR FORCE RESEARCH LABORATORY
Space Vehicles Directorate
3550 Aberdeen Ave SE
AIR FORCE MATERIEL COMMAND
KIRTLAND AIR FORCE BASE, NM 87117-5776

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 is the result of contracted fundamental research which is exempt from public affairs security and policy review in accordance with AFI 61-201, paragraph 2.3.5.1. This report 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-RV-PS-TR-2022-0062 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

//SIGNED//

THOMAS A. LOVELL
Program Manager

//SIGNED//

ANDREW SINCLAIR
Tech Advisor

//SIGNED//

JOHN BEAUCHEMIN
Chief Engineer, Spacecraft Technology Division
Space Vehicles 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.

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
18-03-2022	Final Report	06 Mar 2020 – 18 Mar 2022

4. TITLE AND SUBTITLE
A Multistage Stochastic Optimization Approach for Identifying Stable and Influential Clusters in Randomly Changing Networks

5a. CONTRACT NUMBER
FA9453-20-1-0007

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
61102F

6. AUTHOR(S)
Maciej Rysz

5d. PROJECT NUMBER
3003

5e. TASK NUMBER

5f. WORK UNIT NUMBER
V1T1

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Miami University
501 E. High Street
Oxford, OH 45056

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory
Space Vehicles Directorate
3550 Aberdeen Avenue SE
Kirtland AFB, NM 87117-5776

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

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-RV-PS-TR-2022-0062

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
A Multistage Stochastic Optimization Approach for Identifying Stable and Influential Clusters in Randomly Changing Networks

14. ABSTRACT
The primary task of the proposed research is to develop novel decision models and solution strategies for topological characterization of inter-connected dynamical systems or sensor networks that can maintain functionality, performance and communication in highly uncertain operational settings. We first introduce a model for finding clusters (i.e., groups of nodes) that can control a network. Then, we develop a decision framework for identifying optimal sensor (e.g., satellite) placement locations such that network-wide communications can be ensured and restored/repaired within available resources before and after observing random structural changes. Given a network topology, where the nodes represent potential placement locations, emphasis is put on finding clusters that guarantee inter-sensor communication before and after the random changes occur. Our approach leverages on advances from multistage stochastic and combinatorial optimization to identify clusters that invoke such optimal performance. Results demonstrate that effective applications will facilitate reliable communication network designs evidenced by strong failure tolerances in precarious operational environments. In a supplemental task, we develop a deep neural network learning architecture that uses synthetic aperture radar data for navigation in GPS-denied environments. The described concepts have direct interpretations in, for example, developing localization strategies for deployment of multi-agent UAV systems, satellite networks and its vulnerability analysis, and so on.

15. SUBJECT TERMS
Secure Networks, Optimization, Controls, Communications Encryption

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Thomas A. Lovell
Unclassified	Unclassified	Unclassified	Unlimited	38	**19b. TELEPHONE NUMBER** *(include area code)*

(This Page Intentionally Left Blank)

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

1. SUMMARY

A large body of network optimization and graph-theoretic literature focuses on problems of finding groups (sets) embedded in networks (graphs) with prescribed properties that are of the largest or smallest size, weight, influence, etc. Numerous studies have also addressed problem instances of finding sets that exhibit "resilience", "reliability", and "robustness", relative to random changes in the graph's topology. In Task 1 we introduce a deterministic and stochastic variant of the problem of finding structured sets in graphs that include finding a most controllable set and a set that can be repaired given uncertain structural changes to the network, respectively. In the latter case, an important distinction between our studies and prior work involves the added capability for identifying sets that are guaranteed to be reparable in the sense that their structural property can be restored after the network's topology randomly changes. In other words, the desired structural property is necessarily satisfied both before and after the random changes occur to the graph. We developed an exact branch-and-bound algorithm for solving the model. Its computational performance was compared against solving an equivalent mathematical programming formulation using standard optimization solver software and demonstrated to be far superior.

Another desirable feature to incorporate in a wireless sensor network (WSN), e.g., a satellite network, is the capability to securely transfer information among its components. This requires sophisticated strategies that distribute encryption keys among the network's nodes. To this end, in Task 2 we developed an optimization-based modeling framework using the q-Composite encryption scheme for finding optimal encryption key management policies that produce a desired level of communication security in both deterministic and stochastic networks. The proposed model offers encryption strategies that provide a user-defined level of security while considering the underlying WSN's topology and the limited memory capacities of nodes relative to the size of the key pool. Numerical experiments obtained from solving the models for various network configurations demonstrate the effectiveness and computational efforts. The developed models can be used to obtain optimal encryption strategies in small scale networks.

Finally, in Task 3 we present a novel approach to aid navigation in GPS-denied environments by using a deep neural network-based synthetic aperture radar (SAR) image descriptor. Namely, a novel *deep cosine similarity neural network* (DCSNN) that utilizes a graph-based representation of SAR image patches for training is introduced. The use of cosine similarity, induced by the DCSNN, enabled effective measurements of distances between feature vectors of SAR image patches in a scalable manner. Then, a navigation procedure using SAR image matching, retrieval, and registration that obtains and correlates current coordinates of a vehicle with coordinates retrieved from a database was developed. The methodology was validated and shown to be effective for polarimetric SAR (PolSAR) image data from the Uninhabited Aerial Vehicle Synthetic Aperture Radar (UAVSAR) database.

TASK 1: SUBGRAPH DETECTION MODELS

1.0 INTRODUCTION

A large body of network optimization and graph-theoretic literature focuses on problems of finding groups (sets) embedded in networks (graphs) with prescribed properties that are of the largest or smallest size, weight, influence, etc. [1-4]. Numerous studies have also addressed problem instances of finding sets that exhume "resilience", "reliability", and "robustness", relative to random changes in the graph's topology (see [5-11]). A common approach for finding such sets involves identifying components that collectively satisfy a prescribed structural property before and/or after failures of a network's edges or nodes. Several examples comprise node and link connectedness, maximization of overall algebraic connectivity, network flow control, and prevention of catastrophic cascade failures [8, 12-17]. To investigate the problem of finding sets in graphs whose topology randomly changes, our work [11, 18] introduced modeling frameworks based on the notion of two-stage stochastic programming [19] for finding groups with user-defined structural properties. An important distinction between our studies and prior literature involves the added capability for identifying sets that are guaranteed to be reparable in the sense that their structural property can be restored after the network's topology randomly changes. In other words, the desired property is necessarily satisfied both before and after the random changes occur.

The underlying core deterministic problem involves identifying a largest, smallest, or a most "influential" set of nodes in a graph that collectively possess a specified structural property. Let $G = (V, E)$ represent an undirected *graph* where each *vertex* $i \in V$ is a component of the network, and an *edge* $(i, j) \in E$ defines a connection between vertices i and j. Then, for instance, the problem of finding a set $S \subseteq V$ of vertices with the maximum cardinality, or the *maximum subgraph problem*, takes the form

$$\max_{S \subseteq V} \{|S| : G[S] \text{ satisfies } \Pi\}, \tag{1}$$

where Π represents a user-defined structural property that set S must satisfy (e.g., all set member pairs must share an edge), and $G[S]$ denotes the subgraph of G *induced* by S, i.e., a graph such that any of its vertices i, j are connected by an edge if and only if (i, j) is an edge in G. The structural property Π in (1) is typically chosen to provide a certain level of "robustness" with respect to the connection characteristics between the vertices of set S. In the next subsections we describe the determistic and stochastic variants of problem (1) that include finding a most controllable set and a set that can be repaired given uncertain structural changes to the network, respectively.

2.0 METHODS, ASSUMPTIONS, AND PROCEDURES

2.1 Minimum control energy subgraph detection model

In addition to property Π, a complementary aspect in analyzing complex systems that stems from the applicant's preliminary work [20] involves identifying sets that are "influential". This encompasses methods for quantifying the centrality of groups embedded in graphs [21-23], and represents an important extension associated with the problem of finding a single vertex of highest centrality in graphs. In this work, we define a subgraph as "influential" if it can easily control a given network. In other words, we consider the problem of finding sets of nodes of property Π that can most "easily" control a network. Given a property Π that the selected set must satisfy, the *minimum control energy subgraph detection problem* can be expressed as

$$\min_{S\subseteq V}\{\mathcal{C}(S): G[S] \text{ satisfies } \Pi, \quad \mathcal{F}(S) \leq 0\}, \tag{2}$$

where the *control measure* $\mathcal{C}(S)$ quantifies the control effort of set S, and $\mathcal{F}(S) \leq 0$ indicates that the set S can control the network (i.e., S is controllable). Specifically, let $\mathcal{C}(S)$ represent the *average control energy* of S defined by $\mathcal{C}(S) = tr\left(W_{S,T}^{-1}\right)$, where $W_{S,T}$ is the controllability Gramian given by

$$W_{S,T} := \sum_{\tau=0}^{T-1} A^{\tau} B_S B_S^{\top} (A^{\top})^{\tau}. \tag{3}$$

In the expression above, $A = [a_{ij}]$ is the adjacency matrix, where $a_{ij} = 1$ if edge $(i,j) \in E$, and $a_{ij} = 0$ if edge $(i,j) \notin E$. Then, $S \subseteq V$ is a set of $m \leq |V|$ controllable nodes, $B_K := [e_{k_1}, \dots, e_{k_m}]$ is an input matrix, where e_i is the i-th canonical vector of dimension $|V|$, and T is the total number of time steps.

An important feature of the average control energy that makes it conducive to finding optimal sets satisfying model (2) directly results from the fact that $\mathcal{C}(S)$ is monotonic with respect to subsets [24]. The property described in the next proposition readily follows:

Proposition Given graph $G = (V, E)$, property Π, and $\mathcal{C}(S)$ defined as average control energy, a optimal solution $S^* \subseteq V$ to model (2) is a maximal set.

Corollary 1 The statement in Proposition 1 implies that $\mathcal{C}(\mathrm{S}_1) \leq \mathrm{C}(\mathrm{S}_2)$ whenever $\mathrm{S}_1 \subseteq \mathrm{S}_2 \subseteq \mathrm{V}$. Therefore, adding nodes to a set cannot deteriorate the corresponding average control energy.

A branch-and-bound (BnB) algorithm for solving model (2) that finds sets in graphs which require the lowest average control energy can be constructed by utilizing the described properties. As a demonstrative example, we focus on cases where property Π defines a clique (i.e., a set of nodes that are all interconnected). The algorithm navigates levels, denoted by l, of the branch-and-bound

"tree". Let Q be a partial solution and candidate set R_l be a set of nodes that can individually be added to Q without disrupting the property Π, i.e., the resulting set remains a clique. Initially, at level $l = 0$ the algorithm removes a node from R_0 and adds it to Q. Then, all nodes that would not individually form a clique with Q are removed from R_0, resulting in the candidate set R_1 at the next level $l + 1$. By virtue of Proposition 1 optimal sets Q^* are maximal, and therefore the algorithm continues adding nodes to Q until the candidate set at the corresponding level is empty. Whenever a partial solution is constructed by adding a node to Q from R_l, the algorithm verifies whether Q is controllable, i.e., $\mathcal{F}(Q) \leq 0$, by computing the eigenvalues of $W_{Q,T}$. If the first eigenvalue is nonnegative, then Q is controllable, otherwise the algorithm backtracks by removing the most recently added node to Q and selects the next node (if any) in the candidate set.

To avoid enumerating every possible solution, a bounding condition is applied to determine if a better solution than a previously obtained Q^* (if any) can be found by adding more nodes from R_l to Q. In particular, an upper bound on the average control energy induced by any clique in $Q \cup R_l$ is given by $tr\left(W_{Q\cup R_l,T}^{-1}\right)$. By applying Proposition 1, given an incumbent solution Q^*, adding nodes from R_l to Q could potentially reduce the average control energy only if $tr\left(W_{Q^*}^{-1}\right) > tr\left(W_{Q\cup R_l,T}^{-1}\right)$. Otherwise, the algorithm backtracks to the previous level by removing the most recently added node to Q. In the case when $R_l := \{\emptyset\}$, the partial solution Q is maximal and becomes the incumbent if $tr\left(W_{Q^*}^{-1}\right) > tr\left(W_{Q,T}^{-1}\right)$. Finally, the algorithm terminates once all the nodes in candidate set R_0 have been removed, and the resulting Q^* is an optimal solution to model (2).

2.2 Two-stage stochastic maximum subgraph problem

Cooperative network systems often undergo dynamic structural changes that can inhibit or alter their functional performance. Therefore, a desirable feature that a network should possess is a high "resilience" against failures of components and links through which they interact. One challenge in this context involves identifying *sets* of components whose functionality can be restored or improved in circumstances when the network's topology randomly changes. In this work, we address the problem of identifying sets that perform a function before and after witnessing random changes to the network structure, where the desired function is defined by a user-selected structural property (e.g., connectedness). Clearly, the structural properties of a set may change if components and links of the network fail or appear, thereby potentially preventing the set from performing its intended function (e.g., set members become disconnected and are unable to communicate). It then becomes of interest to identify sets whose structural properties can be restored or repaired on account of the random changes to the underlying network.

Given that a set's structural property can be disrupted when the network (graph) randomly changes, we assume that a set is resilient if its original property can be restored or repaired within available resources. To this end, the model in this study provides a general decision framework for identifying a resilient set in a network, where the "restoration" of its structural property can be implemented by adding and/or removing components (vertices). It is further assumed that th random changes to the "initial" graph arise in the form of failures or formations of links (edges),

and any realization of the random outcome corresponds to a "modified" graph. Then, in each random outcome, the property of a set is restored by adding or removing vertices from it, where the number of such modifications is limited by a repair budget or cost. The adopted objective is to find a *largest* set with a user-defined property in the initial graph whose expected size among all the random realizations is also as large as possible.

In a mathematical context, the described model of selecting and repairing a set in a graph naturally lends itself to a *two-stage stochastic recourse optimization* framework (e.g., see [19]), which is frequently utilized for modeling "present-day" decisions that are "hedged" against uncertain possibilities that may materialize in the future. A key feature is that a *first-stage* decision is made before knowing the actual realization of uncertain factors, while *second-stage* decisions are made after the uncertainties have been realized. Also, the second-stage decisions are influenced by, and must take into account, the adopted first-stage decision. Overall, given an optimal first-stage decision, optimal second-stage decisions are ones that on average perform the best relative to the first-stage decision.

Let E_0 be the set of edges of the first-stage (initial) graph G_0, and suppose they can randomly fail and new edges can randomly appear. Given a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, where Ω is a set of random events, $\mathcal{F}$ is the sigma-algebra, and $\mathbb{P}$ is the probability measure, let G_0 undergo random changes resulting in a transformed second-stage graph $G(\omega) = \big(V, E(\omega)\big)$, $\omega \in \Omega$. By stochastic programming convention, it is assumed that the set Ω is finite, $\Omega = \{\omega_1, \dots, \omega_N\}$, such that the probability of a realization $\mathbb{P}(\omega_k) = p_k > 0$ and $\sum_k p_k = 1$. To ease notation, denote the scenario set by $\mathcal{N} = \{1, \dots, N\}$ and let the second-stage graph corresponding to scenario $k \in \mathcal{N}$ be denoted by G_k, i.e, $G(\omega_k) = G_k$ for $k \in \mathcal{N}$.

It is then of interest to select a *first-stage* Π-set $S_0 \subseteq V$ that is of the largest possible size in G_0, given that S_0 may not satisfy Π after the random edge modifications occur. Therefore, to restore property Π, a *second-stage* recourse action is taken to "repair" S_0 in each graph G_k, $k \in \mathcal{N}$, by adding and/or removing vertices within a predefined *budget limit* B (e.g., B represents limited resources available to make repairs). For scenario $k \in \mathcal{N}$, the resulting "repaired" Π-set is denoted by S_k. Moreover, the addition and removal of vertices is conducted such that the maximum expected size of second-stage solutions S_k, $\forall k \in \mathcal{N}$, is maximized. Then, the graph-theoretic representation of finding a largest first-stage Π-set such that the expected size of second-stage Π-set is also maximized, or the *two-stage stochastic maximum subgraph problem*, takes the form:

$$\begin{aligned} \max_{S_0 \subseteq V} \quad & |S_0| + \textstyle\sum_{k \in \mathcal{N}} p_k \; \mathcal{P}_k(S_0) \\ \text{s. t.} \quad & G_0[S_0] \;\; \text{satisfies} \;\; \Pi, \end{aligned} \tag{4}$$

where the second-stage function $\mathcal{P}_k$ represents the problem of finding a largest second-stage set in G_k, $k \in \mathcal{N}$, given a first-stage solution S_0:

$$\mathcal{P}_k(S_0) = \max\{|S_k| : G_k[S_k] \;\; \text{satisfies} \;\; \Pi, \; h(S_0, S_k) \leq B\}. \tag{5}$$

The function $h(S_0, S_k)$ represents the cost of selecting S_0 in G_0 and S_k in G_k, and should be constructed to also reflect the cost of modifying/repairing S_0 in order to obtain S_k. Although we assume that the selection and modification process is limited by budget B, the methods proposed in this study can be easily generalized to consider a random budget that can vary among scenarios in set $\mathcal{N}$. Also, to avoid trivial instances of (4), it is assumed that B is not overly restrictive so as to enable the selection of at least one vertex in $G_0, \dots, G_N$; and is sufficiently restrictive to prevent the objective value from simply being given by the expected size of the largest Π-sets in $G_0, \dots, G_N$.

2.2.1 The two-stage stochastic maximum *s*-club problem.

To demonstrate applicability of the proposed algorithm, we let the structural property Π of a selected set be defined by a diameter-based clique relaxation known as a *s*-club [25], which guarantees direct or indirect connectedness between all pairs of vertices in the set. Specifically, given an integer *s*, the distance between any pair of vertices of an *s*-club is no larger than *s* in the (sub)graph induced by its vertices. Further, we apply a repair cost structure that is more conducive to real-life applications by imposing non-uniform repair budgets when restoring the structural property of the selected set.

To define the cost function $h(S_0, S_k)$, $k \in \mathcal{N}$, let the *first-stage cost* $c_i \geq 0$ be the cost of selecting $i \in V$ in G_0, and the *second-stage cost* $d_{ik} \geq 0$ be the cost of selecting i in G_k. Let $b_{ik} > 0$ represent a modification to the first-stage and second-stage costs that is associated with selecting node i in both G_0 and G_k. Then, the cost of selecting a first-stage set S_0 in G_0 and second-stage set S_k in G_k is defined as

$$h(S_0, S_k) = \sum_{i \in S_0} c_i + \sum_{i \in S_k} d_{ik} - \sum_{i \in S_0 \cap S_k} b_{ik}. \tag{6}$$

Then, the *two-stage stochastic maximum s-club problem* takes the following form:

$$\begin{aligned} \max_{S_0 \subseteq V} \quad & |S_0| + \sum_{k \in \mathcal{N}} p_k \, \mathcal{P}_k(S_0) \\ \text{s. t.} \quad & G_k[S_k] \text{ is an } s\text{-club}, \quad \forall k \in \{0\} \cup \mathcal{N}, \end{aligned} \tag{7}$$

where,

$$\begin{aligned} \mathcal{P}_k(S_0) = \quad & \max\{|S_k| : G_k[S_k] \text{ is an } s\text{-club}, \\ & \sum_{i \in S_0} c_i + \sum_{i \in S_k} d_{ik} - \sum_{i \in S_0 \cap S_k} b_{ik} \leq B\}. \end{aligned} \tag{8}$$

We developed an exact BnB algorithm for solving the problem in (7) relies on a *first-stage BnB algorithm* for identifying feasible first-stage solutions in G_0, and an embedded *second-stage BnB algorithm* for solving (8) that identifies feasible second-stage solutions in G_k, $\forall k \in \mathcal{N}$. Its computational performance was compared against solving an equivalent mathematical programming formulation using standard optimization solver software.

2.2.2 An integer programming representation of the two-stage stochastic maximum *s*-club problem.

To demonstrate the effectiveness of the developed BnB algorithm, we define an equivalent integer programming formulation of (7). Let binary decision variables x_{ik}, $k \in \{0\} \cup \mathcal{N}$, indicate whether node $i \in V$ belongs to a set S_k in G_k:

$$x_{ik} = \begin{cases} 1, & i \in S_k \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

We extend the formulation of the maximum s-club problem proposed in [26], whereby the mixed-integer programming formulation of the two-stage maximum s-club problem takes the form

$$\begin{aligned}
\max \quad & \sum_{i \in V} x_{i0} + \sum_{k \in \mathcal{N}} p_k \left(\sum_{i \in V} x_{ik} \right) \\
\text{s. t.} \quad & y_{ijk}^{(s)} \geq x_{ik} + x_{jk} - 1, \\
& \quad \forall i, j \in V, \quad i \neq j, \ k \in \{0\} \cup \mathcal{N}, \\
& y_{ijk}^{(1)} = 0, \\
& \quad \forall (i,j) \in \overline{E}_k, \quad i \neq j, \ k \in \{0\} \cup \mathcal{N}, \\
& y_{ijk}^{(l)} = y_{ijk}^{(1)}, \\
& \quad \forall (i,j) \in E_k, \quad l \in \{2, \dots, s\}, \ k \in \{0\} \cup \mathcal{N}, \\
& y_{ijk}^{(l)} \leq \sum_{t:(i,t) \in E_k} y_{tjk}^{(l-1)}, \\
& \quad \forall (i,j) \in \overline{E}_k, \quad l \in \{2, \dots, s\}, \ k \in \{0\} \cup \mathcal{N}, \\
& y_{ijk}^{(l)} \leq x_{ik}, \quad y_{ijk}^{(l)} \leq x_{jk}, \quad y_{ijk}^{(l)} = y_{jik}^{(l)}, \\
& \quad \forall i, j \in V, \quad l \in \{1, \dots, s\}, \ k \in \{0\} \cup \mathcal{N}, \\
& \sum_{i \in V} c_i \, x_{i0} + d_{ik} x_{ik} - b_{ik} x_{i0} x_{ik} \leq B, \\
& \quad \forall k \in \mathcal{N}, \\
& x_{ik} \in \{0,1\}, \quad y_{ijk}^{(l)} \in [0,1], \\
& \quad \forall i, j \in V, \quad l \in \{1, \dots, s\}, \ k \in \{0\} \cup \mathcal{N},
\end{aligned} \tag{10}$$

where $\overline{E}_k$ represents the set of all complement edges of graph G_k. Appropriate mathematical programming solvers can be used to solve formulation (10)

3.0 RESULTS AND DISCUSSION

Computational experiments on various graph configurations were conducted to demonstrate the effectiveness of the proposed BnB algorithm for solving problem (7). All tables demonstrating results are presented in Appendix A. Randomly generated Erdös-Rényi graphs of orders $|V| = 10, 25, 50, 75, 100$ and various densities were considered for distance thresholds of $s = 2,3$. Additionally, experiments were performed on various real-life graph instances from the DIMACS library and Network Repository [27].

The budget limit value was set to $B = \epsilon \cdot \mathrm{E}[\omega(G)]$, where the term

$$\mathrm{E}[\omega(G)] = \frac{2}{\ln(1/d)} \ln|V(G^s)| + o(\ln|V(G^s)|), \tag{11}$$

represents the expected size of the maximum clique in a uniform random (power) graph of G [28]. To maintain a restrictive budget, we let $\epsilon = 0.1(s-1)$. In order to guarantee the existence of a solution with non-zero cardinality, we additionally require that there exists at least one vertex $i \in V$ such that $c_i \leq B$. First-stage costs $c_i \in (0,1)$, $\forall i \in V$, and second-stage costs coefficients $d_{ik} \in (1,2)$, $\forall i \in V$, $\forall k \in \mathcal{N}$, were assigned such that Assumptions 1–2 are satisfied.

The developed branch-and bound algorithm was implemented in C++, and CPLEX 12.5 mixed-integer programming solver was utilized to solve problem (8). The computation hardware included an Inter Xeon 3.50GHz PC with 64GB of RAM, and a Windows 7 64-bit environment. A computational time limit of 3600 seconds was imposed and the symbol "|" in the tables below was used to indicate that the time limit was exceeded.

Tables 1-5 demonstrate computational times and the best feasible objective values found within the time limit. Due to a general inability of the CPLEX solver to obtain a feasible solution within the time limit when solving problem (10), Table 1 presents small-scale Erdös-Rényi graphs with $V = 10$ and $N = 5$ for which the solver either found an optimal solution or a feasible solution. In these cases, the second stage graphs were generated by allowing the edges of G_0 to exclusively fail with a probability of 0.5. Note that CPLEX found a feasible solution that produced an objective value equivalent to the optimal solution for several instances where the time limit was exceeded. By examining cases with $s = 2$, it becomes clear that the computational time required by the CPLEX solver drastically increases as the graph's density increases. A similar argument can be made for cases with $s = 3$. Overall, the proposed BnB algorithm found an optimal solution for all the listed instances, and improved computation times by at least four orders of magnitude.

For all remaining experiments, the CPLEX solver failed to produce feasible solutions within the time limit. Therefore, we only report results generated by the developed BnB algorithm. In Table 2, we consider Erdös-Rényi graphs with $V = 25,50,75,100$ and of various densities (listed in

increasing order according to the number of edges.) For each graph configuration, the number of second-stage graphs considered were $N = 50$ and 75. The second-stage graphs were generated by allowing edges and complement edges of G_0 to fail and appear with a probability of $D(G_0)$, respectively. For any given graph size (e.g., $|V| = 50$), it becomes clear that the time required to solve the instances increases with the density. This stems from the fact that a larger number of feasible s-clubs generally exist when links are added to a graph. Also, for a given configuration of $|V|$, $|E|$ and N, the computation time also generally increases with s.

Table 3 presents the results obtained from the BnB algorithm for various real-life graphs. Although analogous observations about computation times can be made about the relative improvement over CPLEX, there do exist cases which posed challenges for the BnB algorithm. For example, the BnB algorithm was not able to find an optimal solution for the graph "chesapeake", whereas optimal solutions were obtained for graphs of similar densities (e.g., "hi-tech" and "London-gang") when $s = 2$ and/or 3.

In Tables 2-3, notice that there are several instances where the BnB algorithm had difficulty identifying feasible solutions that produce large objective values (e.g., see graphs "lesmis" and "football" in Table 3 when $N = 50$ and $s = 3$). In such cases, it was observed that the algorithm failed to explore a larger number of feasible solutions. Empirical observations suggest that this resulted from a lax bounding condition $\omega(\cdot)$ when considering third order power graphs of the first-stage and second-stage graphs. Namely, the power graphs corresponding to these instances exhibited very high densities, and therefore generally form s-clubs in their entirety when $s = 3$. For example, the average degree of each vertex in power graph G_0^3 of "football" and "lemsis" was 108.6 and 76, respectively. Nevertheless, this tendency is nonrestrictive, as changes in the number of second-stage graphs and cost coefficient data can inhibit such behavior. Indeed, this can be observed for the graph "lesmis", where the case of $N = 50$ posed more difficulty (relative to obtaining a higher objective value) than the larger instance with $N = 75$.

Table 4 furnishes results for an Erdös-Rényi graph with $|V| = 50$, $|E| = 85$, $|\mathcal{N}| = 25$ that demonstrate the effects of changes in the budget coefficient ϵ in the range $[0.1,0.8]$. It is evident that the computation time generally increases as ϵ increases. This is consistent with the fact that the bounding condition used weakens for higher values of budget B due to the fact that a larger number of first-stage solutions may potentially be repaired in the second-stage graphs. Hence, a larger number of feasible second-stage solutions must be explored, increasing the size of the corresponding search space.

Finally, Table 5 considers an Erdös-Rényi graph with $|V| = 50$, $|E| = 85$, $|\mathcal{N}| = 25$ and 75, where the $|\mathcal{N}|$ second-stage graphs were constructed by exclusively allowing the edges of G_0 to fail with a prescribed probability. These results demonstrate the impact of edge failure probabilities in the range $[0.1,0.9]$ for cases when $s = 2$. Clearly, since more feasible solutions exist when the second-stage graphs possess a larger number of edges, then smaller probabilities of edge failures generally induce longer solution times.

Overall, the developed BnB performed effectively in the vast majority of the instances. This is particularly emphasized by the fact that the CPLEX solver was not able to obtain nontrivial feasible solutions for any of the listed graphs in Tables 2-3.

4.0 CONCLUSIONS

We considered a two-stage stochastic maximum subgraph problem of finding s-clubs in graphs whose expected size is maximized after the original graph undergoes random structural changes. Focus was placed on finding *s*-clubs whose properties can be restored within a finite repair budget in the presence of random edge failures and appearances. We developed a combinatorial branch-and-bound algorithm that exploits the structural properties of s-clubs and the allowable repair budget. Numerical experiments on various randomly generated graphs and real-life graphs demonstrate that the proposed algorithm can reduce computation times by several orders of magnitude in comparison to a state-of-the-art mathematical programming solver.

Future work will emphasize modeling and theoretical extensions that admit variations in the user-defined structural properties between decision stages. It is also of interest to consider settings where different structural properties are enforced among the sets selected in the second-stage graphs. Publications related to this work are the following:

1. Rysz, M., Mehta, S., (2019) A two-stage stochastic optimization approach for detecting structurally stable clusters in random graphs, *IEEE Transactions on Network Science and Engineering*, 6(4), 671-683.
2. Rysz, M., Mehta, S., The Minimum Control Energy Cluster Detection Problem. (in preparation)

TASK 2: INTEGER PROGRAMMING FORMULATIONS FOR NETWORK COMMUNICATIONS ENCRYPTION

1.0 INTRODUCTION

A desirable feature to incorporate in a wireless sensor network (e.g, satellite network) is the capability to securely transfer information among its components. Achieving this requires the advancement of complex encryption strategies that can guarantee secure communications when one or more network sensors are compromised (e.g., hacked). Although designing communication schemes under this scope does not necessarily require that the underlying network be impervious to any form of attack, it should induce a high tolerance against the interception of system-wide communications whenever a given number of sensors are infiltrated.

Due to the fact that wireless sensor networks (WSNs) are often deployed in volatile operational environments in which nodes may be physically or virtually captured, it is important to develop communication encryption strategies that are resilient in such settings. Establishing secure communications typically involves distributing secret cryptographic *keys* that are used to encrypt and decrypt information transferred between network nodes that are within *proximity* of one another. Particularly, a given pair of nodes within proximity must share a certain number of keys in order to directly communicate in a secure manner. Proximities between sensors may, for example, be determined by various factors like physical distances, geographical obstacles, signal interference, and so on. Then, the problem of distributing keys to sensors, also known as the *key management problem* (KMP) in the cyber security literature, comprises strategies for assigning secret cryptographic keys to nodes so as to enable all pairs of nodes to directly or indirectly communicate. Additionally, in WSNs the key distribution process is constrained by nodes' *capacities* and computation power (if any), thereby limiting the number of keys that they can store and process in a reasonable amount of time, respectively. Overall, well-constructed key distribution schemes can be applied to make communications between the sensors secure from interception and modification.

Many key distribution protocols assume that the topology of the WSN is not known in advance and apply a key pre-distribution scheme before the sensors are deployed. One of the most widely utilized pre-distribution techniques relies on synchronous encryption. Namely, keys are drawn from a *key pool* and assigned to the nodes. Any node's memory capacity, in turn, limits the number of keys it can store to a small fraction of total keys in the key pool. Then, in order for a pair of nodes to communicate, they must share a prescribed number of keys according to a defined scheme. Surveys of key distribution schemes may be found in [29-31].

In the next sections we propose an optimization-based modeling framework using the q-composite scheme (described below) for finding optimal key management policies that produce a desired level of communication security in both deterministic and stochastic networks. The proposed model offers encryption strategies that provide a user-defined level of security while considering the underlying WSN's topology and the limited memory capacities of nodes relative to the size of

the key pool. As a demonstrative example, they employ security requirements that eliminate the overuse of any particular key by limiting the number of times it can be assigned among all the sensor nodes, and the number of times it can be assigned within the neighborhood of any particular node. This prevents overly repetitive establishment of communications via the same key, which is a highly desirable security feature.

2.0 METHODS, ASSUMPTIONS, AND PROCEDURES

2.1 A mathematical programming representation of the q-Composite key distribution scheme

In this subsection, we present an encryption scheme to address the key pre-distribution problem for wireless sensor networks where the topology of the networks is fixed. q -Composite key distribution scheme is employed as the protocol in this study. A special case of the q-Composite key distribution scheme for cases when $q = 1$ is known as the Eschenauer and Gligor's method [32], which is summarized as follows. A key pool comprising a number of keys is randomly generated. Each key in the key pool is associated with a unique identifier. For each sensor node i, a subset of keys is selected from the key pool – the chosen subset is also called the *key ring* of node i. If two sensor nodes within proximity and have at least one common key in their key rings, then they can communicate securely without relying on any intermediary nodes for exchanging information. The procedure that utilizes the key identifiers for discovering common keys between two nodes is referred to as *shared key discovery* [29]. Alternatively, if the key rings of two nodes do not share any common keys, they can (usually) communicate through a procedure called *key path establishment*. In this case, the nodes communicate indirectly through a sequence of secure links, where the sensors at the two endpoints of any given link share a common key. The Figure 1 (left) illustrates the Eschenauer and Gligor's method on a simple network with a predetermined key ring. For instance, the key ring associated with node A is given by $\{6,7,9\}$. Observe that sensor nodes A, B, and C are within proximity of one another, and the key rings corresponding to any given pair share a common key. Hence, each of the three sensors can directly communicate with one another. However, although sensor D is within proximity of sensor B , they cannot communicate directly due to the fact that their key rings do not contain a common key. Consequently, sensor D becomes an isolated node of the network in this case.

As in the Eschenauer and Gligor's method, the q-Composite scheme uses the same procedure for distributing keys among the sensor nodes, for shared key discovery, and for establishing key paths. However, two nodes that are within proximity must have at least q common keys in their

Figure 1. An illustration of Eschenauer and Gligor's scheme (left) and the q-Composite scheme with q = 2

key rings in order to communicate. In this sense, the q-Composite scheme offers a higher level of cryptographic security in that multiple keys may be required to establish inter-sensor communications. Figure 1 (right) illustrates the mechanism of the q-Composite scheme when $q = 2$. Notice that sensors A and B only have one common key, i.e., key 5, thus they are not able to communicate directly since their respective key rings must share at least 2 common keys. However, sensors A and B can communicate indirectly "through" sensor C through the established key path.

2.2 A mathematical programming representation of the q-Composite key distribution scheme for networks with deterministic topology

In this subsection, we discuss the assumptions and objective of the defined q-composite key distribution problem, followed by a corresponding integer programming model. Given a network whose topology is known, each sensor node has a limited storage capacity that restricts the number of keys it can store. To assign keys in a highly secure manner, they should be distributed so as to avoid assigning the same q common keys to more than one node within any given sensor's proximity. Additionally, the overuse of any given key is restricted by limiting the number of times it can be assigned to the nodes. Altogether, the emergent optimization problem involves maximizing the number of sensors that can communicate, all while satisfying the sensors' memory capacities and the required communication security level. We next present an integer program with quadratic constraints that can provide optimal key management policies for the q-Composite key distribution scheme under the defined setting.

Let $G = (V, E)$ be an undirected simple graph where each vertex is a sensor node of the network, and an edge $(i, j) \in E$ indicates that vertices i and j are within proximity of one another. Let the set $V(i) = \{j: (i, j) \in E\}$ contain the neighboring vertices of vertex $i \in V$. Denote K as the set of randomly generated keys that define the key pool, m_k as the amount of memory required to store key $k \in K$, c_i as the memory capacity limit of vertex $i \in V$, and t_k as the maximum number of vertices that key k can be assigned to. Finally, we define a parameter $\theta \in [0,1]$ to adjust the security "level" by regulating the number of times a key can be used for communications by any given vertex.

For a given graph G and a set of keys K, define the following decision variables:

$$x_{ik} = \begin{cases} 1, & \text{if key } k \in K \text{ is assigned to node } i \in V \\ 0, & \text{otherwise,} \end{cases} \tag{12}$$

$$z_{ij} = \begin{cases} 1, & \text{if edge } (i,j) \in E \text{ have } q \text{ common keys} \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

Then, the described q-Composite KMP can be formulated as follows:

$$\max_{x,z} \quad \sum_{(i,j)\in E} z_{ij} \tag{14a}$$

$$\text{s. t.} \quad \sum_{k\in K} m_k\, x_{ik} \le c_i, \quad \forall\, i \in V \tag{14b}$$

$$\sum_{k\in K} x_{ik}\, x_{jk} \ge q z_{ij}, \quad \forall\, (i,j) \in E \tag{14c}$$

$$\sum_{j\in V(i)} x_{ik}\, x_{jk} \le \theta |V(i)| + \alpha, \quad \forall\, i \in V, \forall k \in K \tag{14d}$$

$$\sum_{i\in V} x_{ik} \le t_k, \quad \forall\, k \in K \tag{14e}$$

$$x_{ik} \in \{0,1\}, \quad \forall\, i \in V,\ \forall\, k \in K \tag{14f}$$

$$z_{ij} \in \{0,1\}, \quad \forall\, (i,j) \in E, \tag{14g}$$

where the objective (14a) maximizes the number of vertices that can directly communicate, thereby maximizing the communication effectiveness of the network. Constraints (14b) enforce that the key ring of any given vertex $i \in V$ does not exceed its memory capacity limit c_i. Constraints (14c) determine whether a given vertex pair (i,j) that are within proximity share at least q common keys. Constraint (14d) restricts the number of times that key k can be used by vertex i to encrypt communications with vertices in its neighborhood $V(i)$. The parameter $p \in [0,1]$ represents a percentage, while α is a positive integer that is adjusted to ensure that $p|V(i)| + \alpha$ is greater or equal to 1. Finally, to prevent excessive use of any key $k \in K$, constraints (14e) prevents k from being assigned more than t_k times among all the vertices in V.

2.3 A mathematical programming representation of the q-Composite key distribution scheme for networks with uncertain topology

Key distribution protocols often assume settings where the WSN topology is not known before assigning keys to nodes, where keys are distributed before the nodes are deployed. A broad class of methods that employ such distribution are known as synchronous encryption protocols. In particular, keys assigned to nodes are selected from a *key pool* such that nodes must share a prescribed number of keys in order to communicate. Notice that network-wide communications are observed after deployment and therefore not guaranteed.

Let $\mathcal{N} := \{1, \dots, N\}$ denote the set of scenarios and accordingly define a set of network topology scenarios: $\mathcal{E} := \{E_1, \dots, E_N\}$. We define the following feasible set

$$\mathcal{X} := \left\{ x : \begin{array}{l} \sum_{i \in V} x_{ik} \le t_k \quad \forall\, k \in K \\ x_{ik} \in \{0,1\} \quad \forall\, i \in V,\ \forall\, k \in K \\ \sum_{j \in N(i)} x_{ik}\, x_{jk} \le \theta |V(i)| + \alpha \quad \forall\, i \in V,\ \forall\, k \in K \end{array} \right\}. \tag{15}$$

Then we formulate the following stochastic programming problem

$$\begin{aligned} \max_{x} \quad & \sum_{n \in \mathcal{N}} p_n\, Q_n(x) \\ \text{s. t.} \quad & x \in \mathcal{X}, \end{aligned} \tag{16}$$

where an optimal key pre-distribution solution is searched to maximize the expected communication connectivity of the network and p_n denotes by the probability of scenario n occurring, i.e., $\sum_{n \in \mathcal{N}} p_n = 1$. In particular, the communication connectivity of the network in the nth scenario $Q_n(x)$ is defined as follows:

$$Q_n(x) := \max \quad \sum_{(i,j) \in E_n} z_{ij} \tag{17a}$$

$$\text{s. t.} \quad \sum_{i \in V} f_{si} = |V| - 1 \tag{17b}$$

$$\sum_{j|(j,i) \in E_n} f_{ij} - \sum_{j|(i,j) \in E_n} f_{ji} = 1 \quad \forall\, i \in V \tag{17c}$$

$$f_{ij} \le |V| z_{ij} \quad \forall\, (i,j) \in E_n \tag{17d}$$

$$f_{ij} \ge 0 \quad \forall\, (i,j) \in E'_n \tag{17e}$$

$$\sum_{k \in K} y^k_{(i,j)} \ge z_{ij}, \quad \forall\, (i,j) \in E_n \tag{17f}$$

$$z_{ij} \in \{0,1\} \quad \forall\, (i,j) \in E, \tag{17g}$$

$$\sum_{j \in \mathcal{V}(i)} y^k_{(i,j)} \le \theta |\mathcal{V}(i)| + \alpha \quad \forall\, i \in V,\ \forall\, k \in K \tag{17h}$$

$$y^k_{(i,j)} \le x_{ik},\ y^k_{(i,j)} \le x_{jk} \quad \forall\, (i,j) \in E_n,\ \forall k \in K \tag{17i}$$

$$y^k_{(i,j)} \ge x_{ik} + x_{jk} - 1 \quad \forall\, (i,j) \in E_n,\ \forall\, k \in K \tag{17j}$$

$$y^k_{(i,j)} \in \{0,1\} \quad \forall\, (i,j) \in E_n,\ \forall\, k \in K. \tag{17k}$$

where $E'_n := E_n \cup \{(s,i) \mid \forall\, i \in V\}$. Note that constraints (17f) and (17h) imply that the connectivity establishment on the link $(i,j) \in E_n$ requires q common keys shared between nodes i and j, that constraints (17b) – (17e) enforce that all nodes on the graph are connected, and that the objective (17a) is to maximize the connected links on the network topology E_n.

3.0 RESULTS AND DISCUSSION

Numerical experiments analyzing the effectiveness and solution performance of the proposed key management models were conducted. All tables demonstrating results are presented in Appendix A. Problem (14) was solved for cases when $q = 1$ for randomly generated Erdős-Rényi graphs of orders $|V| = 10, 30, 50, 100$ with average densities in the range $[0.04, 0.5]$. For any graph configuration of order $|V|$ and density d, one hundred instances were generated and solved. Depending on the particular graph instance, the memory capacity limit of vertices was set to $c_i = 5, 6, 7, 8$, and the total number of times a given key $k \in K$ could be assigned was $t_k = 3, 4, 5$. The larger values of c_i and t_k were assigned when solving graph configurations of the larger orders. Throughout the numerical experiments, we let $m_k = 1, \forall k \in K$.

For each configuration, the average computational times of the instances that solved within the time limit were reported. Whenever an optimal solution was not obtained for all the instances of a configuration, the average computational time was reported for the instances that were solved within the time limit, while the average optimality gap was reported for the remaining instances.

Eight graph configurations with various combinations of $|V|$ in the range $[10, 100]$ and d in the range $[0.03, 0.5]$ were considered for $q = 1$. For each configuration, Table 6 lists values for $|V|$, d and parameters $|K|$, q, θ, c_i, t_k; and the average solution times and/or average optimality gaps. The computational time limit for each instance was set to 7200 seconds. The symbol "–" was used to report the solution time for configurations where the computational time limit was exceeded for all one hundred instances. For a given order $|V|$, Table 6 demonstrates that the solution time significantly increases as the graph density increases.

For the stochastic problem (16), we considered the nine problem instances and their configurations listed in Table 7. For demonstrative purposes, we show the computational statistics for q = 1 and 10 scenarios. The scenarios corresponded to realizations – generated randomly based on Erdős-Rényi model with probability equal to d – of unknown topology for the particular problem. The table shows the number of solved instances, where "TTO" denotes their average time to optimality, "TL" denotes count of problem instances stopped due to time computational limit, "TL, inf" denotes count of infeasible problem instances stopped due to time limit, and "avg. gap (timeout)" denotes average optimality gap among the feasible solutions of the problems that stopped due to the time limit.

4.0 CONCLUSIONS

We introduced an integer programming approach for the key management problem in wireless sensor networks in setting where the network topologies may or may not change. Focus was put on the q-Composite key distribution scheme for assigning secret keys to the network nodes. Numerical experiments for cases with q = 1, 2 were conducted to show the applicability and computation effort necessary to solve the developed model. The results suggest that the model is effective for small-scale networks. Our subsequent work will focus on developing solution

algorithms for identification of optimal key allocation in larger networks in both, deterministic settings where the topology is known in advance, and stochastic settings presented above where it is uncertain. Publications related to this research are the following:

1. Xu, G., Semenov, A., Rysz, M. (2020), An integer programming approach for the key management problem in wireless sensor networks, *Optimization Letters*, 14(5):1037 - 1051.

2. Rysz, M., Xu, G., Semenov, A., (2021) Pre-distribution Schemes for Key Management in a Stochastic Environment, *Journal of Combinatorial Optimization*. (under review)

TASK 3: SYNTHETIC-APERTURE RADAR IMAGE-BASED POSITIONING IN GPS-DENIED ENVIRONMENTS USING DEEP COSINE SIMILARITY NEURAL NETWORKS

1.0 INTRODUCTION

Unmanned aerial vehicles (UAVs) have become an integral part of reconnaissance and defense applications, and routinely operate in hostile and uncertain environments. UAV flight maneuvering and navigation can be autonomously controlled and often relies on an embedded global positioning system (GPS) receiver. Although GPS provides position, navigation and timing (PNT) information, in GPS-denied settings it is necessary to obtain analogous spatial and temporal awareness via other mechanisms. Such settings may include, but are not limited to, operations in areas with GPS jamming devices, interference and outages. Indeed, experiments have demonstrated that even a low-power jamming device can interfere with GPS signals, resulting in possible denial of GPS service over large areas [33]. Additionally, attackers can control a maritime surface vessel by broadcasting counterfeit GPS signals in order to manipulate a target receiver's position, velocity, or time [34].

Numerous studies in the literature have considered aided navigation in GPS-denied environments. A classical approach relies on Dead Reckoning (DR), which involves estimating position based on a previously determined position integrated with velocity or acceleration. Although it has been shown to work effectively in the absence of GPS signal, a major drawback stems from the fact that it accumulates position errors over time.

Vision-aided navigation may be a promising alternative that has been widely studied for a decade [35–37]. Visual Odometry (VO), which was termed by Nister *et al.* [37], estimates position and orientation by analyzing the sequence of images. Namely, VO estimates the UAV's current position with respect to a previously acquired position by accumulating inter-frame translation and rotation. VO can also be combined with the Simultaneous Localization and Mapping (SLAM) technique [38, 39] as well as several other fusing methods including filtering methods such as extended Kalman filter [36, 40] and State-Dependent Riccati Equation nonlinear filter [41]. Furthermore, there exist previous works that fused measured information from inertial measurement unit (IMU) [36, 40, 41], and on-board cameras [35, 39]. Another subject of emphasis

within the scope considered image registration aided navigation [42, 43]. Mo Shan *et al.* [42] proposed a method that used image feature extraction via Histograms of Oriented Gradients (HOG), which demonstrated promising results on image registration on Google Maps.

More recent image matching and registration techniques have exploited capabilities of deep neural networks (DNNs) [44, 45, 46]. Convolutional neural networks (CNNs) have been widely used for mapping complicated images to "simpler" feature vectors. The feature vectors, which are also called *global descriptors* of images, are used to compare and retrieve similar images from a database. Also, faster and more scalable CNN inference allows for large-scale image retrieval tasks [47] that would otherwise be difficult with conventional image descriptors such as scale-invariant feature transform (SIFT) or learned invariant feature transformation (LIFT) [48].

In the next subsection we present a novel approach to aid navigation in GPS-denied environments by using a CNN-based SAR image descriptor. To this end, a novel CNN-based image descriptor, the *deep cosine similarity neural network* (DCSNN), that utilizes a graph-based representation of SAR image patches for training is introduced. The use of cosine similarity, induced by the DCSNN, enabled effective measurements of distances between feature vectors of SAR image patches in a scalable manner. Then, a navigation procedure using SAR image matching, retrieval, and registration that obtains and correlates current coordinates of a vehicle with coordinates retrieved from a database was developed. Finally, the methodology was validated and shown to be effective for polarimetric SAR (PolSAR) image data from UAVSAR.

2.0 METHODS, ASSUMPTIONS, AND PROCEDURES

CNN-based descriptors have been effectively used in numerous image analysis and retrieval applications. Structuring our methodology accordingly, the primary goal of the DCSNN model is to efficiently construct a ``simple'' descriptive vector of a given SAR image. The descriptive feature vector of an inquiry patch image can then be compared against feature vectors of patch images stored in a database, thereby, retrieving similar patches along with their location coordinates (e.g., latitude and longitude).

To enhance the performance of tasks that use SAR data, a new CNN-based descriptor can be trained on the SAR data by using the pretrained CNN-based descriptor as an initialization for the new descriptor, which is also known as *fine-tuning*. To "fine-tune" the neural network model in a supervised manner, we use the adjacency matrix $\mathrm{A} \in \mathbb{R}^{N \times N}$, where N is the number of nodes (image patches). Given patches $(i, j) \in \{1, ..., N\}$, let element a_{ij} of matrix A correspond to the edge value defined as the common pixel area ratio between patches i and j. Formally, the edge values, a_{ij}, $\forall (i, j) \in \{1, ..., N\}$, measure the similarities between patches as,

$$a_{ij} = \max\left(0, \frac{2 \cdot Area}{Area_i + Area_j}\right), \forall i, j \in \{1, \ldots, N\}, \tag{18}$$

where Area_i and Area_j are pixel areas of patches i and j, respectively; and Area_{ij} is the intersection of pixel area between patches i and j.

Define the DCSNN model as a CNN-based mapping f_θ parameterized by θ, where the parameters θ are learned during fine-tuning. Let $d \in \mathbb{R}^l$ be a feature vector of feature length l that is obtained from the DCSNN as $d = f_\theta(x)$, where x is an image from the image set X. Clearly, to exclusively use the feature vector d for comparing patch images during the retrieval process, it is required that it be sufficiently ``compact" yet descriptive of the image x. For image patches stored in the database, we additionally construct an *anchor matrix* $D \in \mathbb{R}^{l \times N}$ such that the i$^{\text{th}}$ column corresponds to the feature vector $d^{(i)}$ of the patch $x^{(i)} \in X$.

We define the loss function, $L^{(i)}$, of the DCSNN as follows. For each patch $x^{(i)}$ in the database, $L^{(i)}$ consists of a *cross-entropy loss* $L_{ce}^{(i)}$ and a *regularization loss* $L_{reg}^{(i)}$:

$$L^{(i)} = L_{ce}^{(i)} + L_{reg}^{(i)} \tag{19}$$

where $\lambda > 0$ is a regularization factor. Let a_i be the i$^{\text{th}}$ row vector of the adjacency matrix A, then the cross-entropy loss takes the form,

$$\begin{gathered} L_{ce}^{(i)} = -\left(a_i^T \log \Omega^{(i)} + \left(1 - a_i^T\right) \log\left(1 - \Omega^{(i)}\right)\right), \\ \text{where } \Omega^{(i)} = \sigma\left(\frac{D^T d^{(i)}}{s}\right) \end{gathered} \tag{20}$$

and $\sigma(\cdot)$ is an element-wise sigmoid function, i.e., $\sigma(x) = \dfrac{1}{1+\exp^{-x}}$, that forces the dot product of feature vectors to range between 0 and 1. Since the gradient of the sigmoid function $\sigma(\cdot)$ approaches 0 as the value of x nears $\pm\infty$, it results in longer training times. To mitigate this, we impose a *similarity factor* $s > 0$. Observing that an element a_{ij} of the adjacency matrix A must be in the range [0,1], a_{ij} can be interpreted as a probability that the patch i is equivalent to the patch j. Thus, the elements of A can serve as ``ground truth" probabilities, whereas $\Omega^{(i)}$ is the predicted probability that patch i is similar to other patches.

Observe that the j^{th} element of $\Omega^{(i)}$, denoted by $\Omega_j^{(i)}$, corresponds to the estimated probability that patch i is similar to patch j. Thus, the higher the value of an element a_{ij} is, the higher the value of $\Omega_j^{(i)}$ that the DCSNN is expected to generate. Accordingly, we apply a regularization loss, $L_{reg}^{(i)}$,

such that the norm of the feature vector produced by the DCSNN is expected to be approximately 1 upon successful fine-tuning. To this effect, the regularization loss considered is defined as,

$$L_{reg}^{(i)} = \left(\left\| d^{(i)} \right\|_2 - 1 \right)^2 \tag{21}$$

After training, by using ``regularized" feature vectors from the DCSNN, it is expected that the dot products of the feature vectors are consistent with their cosine similarities. Thus, the cosine similarity between feature vectors generated by the model measures the extent of adjacency of the image patches. This fact motivated the adopted name ``DCSNN".

Finally, by averaging over the N patches, the total loss function takes the form

$$L = L_{ce} + \lambda L_{reg} = \frac{1}{N} \sum_{i=1}^{N} \left(L_{ce}^{(i)} + L_{reg}^{(i)} \right). \tag{22}$$

A stochastic gradient descent algorithm is then used to minimize the above expression.

3.0 RESULTS AND DISCUSSION

The performance of the developed approach was explored in the context of navigation. An affine transformation was used to estimate the coordinates of a given inquiry SAR image. Table 8 in Appendix A furnishes the estimated coordinates produced by our approach for the inquiry SAR image examples shown in Figure 2, which provides examples of the retrieved SAR patches before and after reranking processes. In Figure 2, the first column represents examples of inquiry SAR patches. The first two rows ((a) and (b)) are from Hayward Fault Zone PolSAR and the latter two rows ((c) and (d)) are from Yukon-Kuskokwim Delta PolSAR data. A patch surrounded by a green box indicates it is correctly retrieved, whereas a red box indicates that it is incorrectly retrieved. A DCSNN with the well-known AlexNet architecture, feature length of $l = 120$, and the SAR-SIFT methods for generating keypoints, were used for this experiment. The “Actual Coordinates” and “Estimated Coordinates” in Table 8 represent the latitude and longitude pairs of ground truth and prediction, respectively. Errors were computed using the geodesic distance between actual and estimate coordinates. Observe that the errors are quite small, the largest of which was only $5.7m$. This suggests that the proposed technique can be effective for navigational tasks.

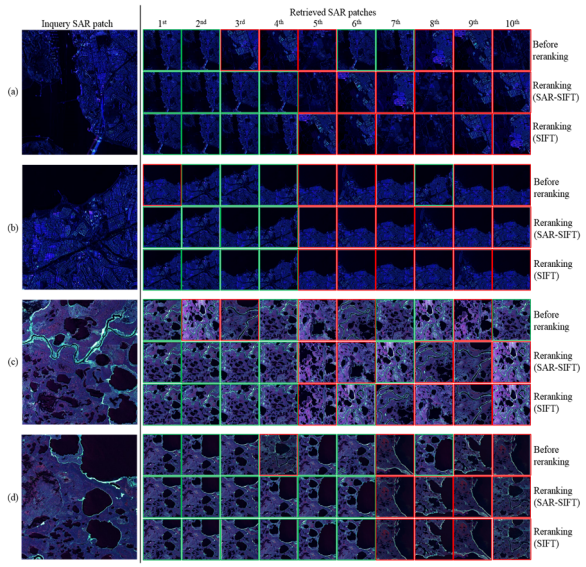

Figure 2. Examples of the retrieved SAR patches before and after reranking processes

4.0 CONCLUSIONS

This work introduced a navigational approach that relies on SAR image processing via deep neural networks to enable effective image matching, retrieval, and registration. We developed a deep neural network-based SAR descriptor, the DCSNN, and a fine-tuning procedure that was used to describe a SAR image with a "simple" feature vector. It was also demonstrated that our method can increase the performance of the image retrieval process and associated navigation tasks on PolSAR data. Publications related to this work include:

1. Park, S., Rysz, M., Fair, K., Pardalos, P., (2021) SAR Image-based Positioning in GPS-denied Environments using Deep Cosine Similarity Neural Networks, *Inverse Problems & Imaging*, 15(4):763-785.

2. Tsokas, A., Rysz, M., Dipple, K, Pardalos, P., (2021) SAR Data Applications in Earth Observation: an Overview, *Expert Systems with Applications*. (to appear)

REFERENCES

[1] P M Pardalos and J Xue. The maximum clique problem. *Journal of Global Optimization*, 4:301–328, 1994.

[2] F M Pajouh and B Balasundaram. On inclusionwise maximal and maximum cardinality *k*-clubs in graphs. *Discrete Optimization*, 9(2):84 – 97, 2012.

[3] R Bar-Yehuda and S Even. A linear-time approximation algorithm for the weighted vertex cover problem. *Journal of Algorithms*, 2(2):198–203, 1981.

[4] H N Gabow, Z Galil, T Spencer, and R E Tarjan. Efficient algorithms for finding minimum spanning trees in undirected and directed graphs. *Combinatorica*,6(2):109–122, Jun 1986.

[5] A Veremyev and V Boginski. Identifying large robust network clusters via new compact formulations of maximum k-club problems. *European Journal of Operational Research*, 218(2):316 – 326, 2012.

[6] JW Wang and LL Rong. Robustness of the western united states power grid under edge attack strategies due to cascading failures. *Safety Science*, 49(6):807 – 812, 2011.

[7] M Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(2):298–305, 1973.

[8] A Ghosh and S Boyd. Growing well-connected graphs. In *Decision and Control, 2006 45th IEEE Conference on*, pages 6605–6611, Dec 2006.

[9] M Rysz, M Mirghorbani, P Krokhmal, and E L Pasiliao. On risk-averse maximum weighted subgraph problems. *Journal of Combinatorial Optimization*, 28(1):167–185, 2014.

[10] M Rysz, F M Pajouh, P Krokhmal, and E L Pasiliao. Identifying risk-averse low-diameter clusters in graphs with stochastic vertex weights. *Annals of Operations Research*, 262(1):89–108, Mar 2018.

[11] M Rysz, P Krokhmal, and E L Pasiliao. Detecting resilient structures in stochastic networks: A two-stage stochastic optimization approach. *Networks*, 69(2):189–204, 2017.

[12] T Kolokolnikov, B Osting, and J VonBrecht. Algebraic connectivity of Erdos-Renyi graphs near the connectivity threshold. *Manuscript in preparation*, 2014.

[13] K Ogiwara, T Fukami, and N Takahashi. Maximizing algebraic connectivity in the space of graphs with fixed number of vertices and edges. *IEEE Transactions on Control of Network Systems*, 99:1–1, 2015.

[14] H Shakeri, N Albin, F D Sahneh, P Poggi-Corradini, and C Scoglio. Maximizing algebraic connectivity in interconnected networks. *Physical Review E*, 93(3):030301, 2016.

[15] V L Boginski, C W Commander, and T Turko. Polynomial-time identification of robust network flows under uncertain arc failures. *Optimization Letters*, 3(3):461–473, 2009.

[16] I Bomze, M Chimani, M Ju¨nger, I Ljubi´c, P Mutzel, and B Zey. Solving two-stage stochastic steiner tree problems by two-stage branch-and-cut. In Otfried Cheong, Kyung-Yong Chwa, and Kunsoo Park, editors, *Algorithms and Computation*, volume 6506 of *Lecture Notes in Computer Science*, pages 427–439. Springer Berlin Heidelberg, 2010.

[17] S V Buldyrev, R Parshani, G Paul, H E Stanley, and S Havlin. Catastrophic cascade of failures in interdependent networks. *Nature*, 464(7291):1025–1028, 2010.

[18] M Rysz and S S Mehta. A two-stage stochastic optimization approach for detecting structurally stable clusters in random graphs. *IEEE Transactions on Network Science and Engineering*,2017, under review - second round.

[19] J R Birge and F Louveaux. *Introduction to Stochastic Programming*. Springer Publishing Company, Incorporated, 2nd edition, 2011.

[20] M Rysz, E L Pasiliao, and F Mi Pajouh. Finding clique clusters with the highest betweenness centrality. *European Journal of Operations Research (accepted)*, 2017.

[21] M G Everett and S P Borgatti. The centrality of groups and classes. *Journal of Mathematical Sociology*, 23(3):181–201, 1999.

[22] M G Everett and S P Borgatti. *Extending Centrality*, pages 57–76. Cambridge university press, 2005.

[23] C Ni, C Sugimoto, and J Jiang. Degree, closeness, and betweenness: Application of group centrality measurements to explore macro-disciplinary evolution diachronically. In *Proceedings of ISSI*, pages 1–13, 2011.

[24] T H Summers, F L Cortesi, and J Lygeros. On submodularity and controllability in complex dynamical networks. *IEEE Transactions on Control of Network Systems*, 3(1):91–101, 2015.

[25] B Balasundaram, S Butenko, and S Trukhanov. Novel approaches for analyzing biological networks. *Journal of Combinatorial Optimization*, 10(1):23–39, 2005.

[26] A Veremyev, O A Prokopyev, and E L Pasiliao. Critical nodes for distancebased connectivity and related problems in graphs. *Networks*, 66(3):170–195, 2015.

[27] R A Rossi and N K Ahmed. The network data repository with interactive graph analytics and visualization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[28] P Erdos and A Renyi. On the evolution of random graphs. 5:17–61, 1960.

[29] CY Chen and HC Chao. A survey of key distribution in wireless sensor networks. *Security Comm. Networks*, 7(12):2495–2508, December 2014.

[30] S Ruj, A Nayak, and I Stojmenovic. Pairwise and Triple Key Distribution in Wireless Sensor Networks with Applications. *IEEE Transactions on Computers*, 62(11):2224–2237, November 2013.

[31] M A Simplıcio, Jr., P S L M Barreto, C B Margi, and T C M B Carvalho. A survey on key management mechanisms for distributed wireless sensor networks. *Computational Networks*, 54(15):2591–2612, October 2010.

[32] L Eschenauer and V D Gligor. A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, CCS '02, pages 41–47, New York, NY, USA, 2002. ACM.

[33] A Grant, P Williams, N Ward, and S Basker. Gps jamming and the impact on maritime navigation. *The Journal of Navigation*, 62(2):173–187, 2009.

[34] J Bhatti and T E Humphreys. Hostile control of ships via false gps signals: Demonstration and detection. *NAVIGATION, Journal of the Institute of Navigation*, 64(1):51–66, 2017.

[35] DG Sim, RH Park, RC Kim, S U Lee, and IC Kim. Integrated position estimation using aerial image sequences. *IEEE transactions on pattern analysis and machine intelligence*, 24(1):1–18, 2002.

[36] S Zhao, F Lin, K Peng, B Chen, and T Lee. Homography-based vision-aided inertial navigation of uavs in unknown environments. In *AIAA Guidance, Navigation, and Control Conference*, page 5033, 2012.

[37] D Nister, O Naroditsky, and J Bergen. Visual odometry. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages 1-1. IEEE, 2004.

[38] P Williams and M Crump. All-source navigation for enhancing uav operations in gps denied environments. In *Proceedings of the 28th International Congress of the Aeronautical Sciences*, 2012.

[39] F Caballero, L Merino, J Ferruz, and A Ollero. Vision-based odometry and slam for medium and high altitude flying uavs. *Journal of Intelligent and Robotic Systems*, 54(1):137–161, 2009.

[40] G Balamurugan, J Valarmathi, and V Naidu. Survey on uav navigation in gps denied environments. In *2016 International conference on signal processing, communication, power and embedded system (SCOPES)*, pages 198–204. IEEE, 2016.

[41] ANemra and N Aouf. Robust ins/gps sensor fusion for uav localization using sdre nonlinear filtering. *IEEE Sensors Journal*, 10(4):789–798, 2010.

[42] M Shan, F Wang, F Lin, Z Gao, Y Z Tang, and B M Chen. Google map aided visual navigation for uavs in gps-denied environment. In *2015 IEEE international conference on robotics and biomimetics (ROBIO)*, pages 114–119. IEEE, 2015.

[43] D O Nitti, F Bovenga, M T Chiaradia, M Greco, and G Pinelli. Feasibility of using synthetic aperture radar to aid uav navigation. *Sensors*, 15(8):18334–18359, 2015.

[44] A Babenko, A Slesarev, A Chigorin, and V Lempitsky. Neural codes for image retrieval. In *European conference on computer vision*, pages 584–599. Springer, 2014.

[45] G Tolias, R Sicre, and H J´egou. Particular object retrieval with integral max-pooling of cnn activations. arxiv 2015. *arXiv preprint arXiv:1511.05879*, 2015.

[46] J YH Ng, F Yang, and L S Davis. Exploiting local features from deep networks for image retrieval. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 53–61, 2015.

[47] H Noh, A Araujo, J Sim, T Weyand, and B Han. Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE international conference on computer vision*, pages 3456–3465, 2017.

[48] K M Yi, E Trulls, V Lepetit, and Pl Fua. Lift: Learned invariant feature transform. In *European conference on computer vision*, pages 467–483. Springer, 2016.

APPENDIX A

LIST OF TABLES

Table 1. Solution times (in seconds) and objective values for problem (8) on Erdos-Renyi graphs The symbol "—" indicates that the time limit of 3600 seconds was exceeded

			s =2				s =3			
Erdos-Renyi Graphs			CPLEX		BnB		CPLEX		BnB	
$\|V\|$	$\|E\|$	|N|	Time(s)	Objective	Time(s)	Objective	Time(s)	Objective	Time(s)	Objective
10	13	5	40.19	3.2	0.001	3.2	405.22	4.6	0.005	4.6
10	14	5	44.27	3.4	0.001	3.4	—	4.8	0.008	4.8
10	18	5	1198.27	4.6	0.001	4.6	—	5.8	0.016	6.6
10	19	5	—	5.2	0.016	5.2	—	7.0	0.093	8.6
10	20	5	3587.15	6.8	0.019	6.8	—	10.6	0.046	10.6
10	24	5	—	3.6	0.008	7.2	—	10.6	0.01	11

Table 2. Solution times (in seconds) and objective values for problem (8) on Erdos-Renyi graphs The symbol "—" indicates that the time limit of 3600 seconds was exceeded

Erdös-Rényi Graphs			$s = 2$		$s = 3$	
$\|V\|$	$\|E\|$	$\|\mathcal{N}\|$	Time (s)	Objective	Time (s)	Objective
25	27	50	0.11	5.8	2.11	7.7
		75	0.20	2.0	0.94	3.9
25	34	50	0.16	3.9	1.72	3.9
		75	0.14	5.8	3.09	7.7
25	49	50	0.94	3.9	12.96	7.6
		75	1.48	5.8	25.03	11.7
25	70	50	4.24	3.8	99.04	9.7
		75	4.01	3.8	55.19	9.5
25	70	50	3.67	5.8	102.40	11.6
		75	10.17	9.6	194.60	17.7
50	71	50	0.34	5.9	13.34	7.8
		75	1.17	4.0	21.78	5.9
50	118	50	2.82	5.7	277.99	7.8
		75	7.85	7.8	1076.18	13.7
50	175	50	48.34	7.8	—	15.7
		75	124.90	5.9	—	13.8
50	232	50	377.59	11.1	—	17.7
		75	986.60	11.1	—	21.7
50	321	50	—	11.2	—	23.9
		75	—	11.3	—	23.8
75	149	50	6.05	6.0	435.10	9.8
		75	4.57	7.8	467.94	11.8
75	310	50	291.49	7.7	—	11.5
		75	618.65	9.7	—	5.8
75	397	50	1344.25	7.8	—	17.7
		75	1412.94	7.8	—	15.6
75	593	50	—	14.9	—	27.9
		75	—	11.4	—	19.8
75	711	50	—	9.2	—	23.7
		75	—	13.1	—	40.0
100	156	50	3.36	7.9	124.32	9.9
		75	5.31	5.9	263.45	7.9
100	251	50	27.95	5.9	—	9.8
		75	9.68	9.9	—	15.6
100	488	50	1352.86	7.7	—	2.0
		75	2255.26	9.6	—	7.7
100	761	50	—	16.9	—	3.8
		75	—	13.5	—	2.0
100	995	50	—	11.1	—	21.6
		75	—	11.2	—	29.9

Table 3. Solution times (in seconds) and objective values for problem (8) on real-life graphs The symbol "—" indicates that the time limit of 3600 seconds was exceeded

				$s = 2$		$s = 3$	
Graph Name	$\|V\|$	$\|E\|$	$\|\mathcal{N}\|$	Time (s)	Objective	Time (s)	Objective
attiro	60	161	50	9.05	5.9	392.33	9.7
			75	6.69	5.9	469.50	9.7
chesapeake	39	170	50	—	13.2	—	27.9
			75	—	12.8	—	21.6
dolphins	62	159	50	8.07	7.8	938.69	9.9
			75	28.02	5.9	1209.76	11.6
football	115	613	50	2861.39	13.9	—	2.0
			75	1117.50	9.9	—	15.7
heroin-dealing-Natarajan	38	84	50	3.69	7.6	172.58	9.7
			75	12.90	5.8	365.92	11.6
hi-tech	36	147	50	10.30	5.6	472.98	11.4
			75	38.36	7.6	1430.08	13.8
Italian gangs	67	114	50	18.15	5.9	995.16	11.9
			75	13.15	3.9	515.53	7.9
jazz	198	2742	50	—	17.2	—	3.0
			75	—	18.9	—	3.0
karate	34	78	50	4.98	5.8	113.48	9.7
			75	16.85	7.6	1405.06	11.9
lesmis	77	254	50	—	13.5	—	2.0
			75	—	9.8	—	19.6
London-gang	54	315	50	3256.41	9.5	—	13.7
			75	2407.19	11.0	—	19.7
Mexican-power	35	117	50	48.61	11.5	—	23.6
			75	21.48	5.7	590.60	13.3
polbooks	105	441	50	567.97	11.7	—	15.6
			75	—	9.8	—	15.8
SanJuanSur-net	75	199	50	5.54	5.9	251.19	10.0
			75	26.18	4.0	856.75	7.9
sawmill	36	62	50	1.01	5.8	13.18	7.8
			75	3.25	3.9	53.84	9.5
siren	44	103	50	62.32	13.5	—	17.8
			75	26.54	5.8	1945.53	15.7

Table 4. Solution times (in seconds) and objective values for problem (8) on Erdos-Renyi graphs with various values of s and a fixed number of vertices, edges, and scenarios The symbol "—" indicates that the time limit of 3600 seconds was exceeded

Erdos-Renyi Graphs				s = 2		s = 3	
$\|V\|$	$\|E\|$	|N|		Time(s)	Objective	Time(s)	Objective
50	85	25	0.1	0.41	3.8	5.48	3.96
50	85	25	0.15	0.73	5.92	21.88	9.6
50	85	25	0.2	0.96	5.96	132.30	11.64
50	85	25	0.25	1.53	7.96	217.78	11.92
50	85	25	0.3	1.97	7.8	305.65	15.72
50	85	25	0.35	2.79	7.92	557.80	17.6
50	85	25	0.4	3.57	7.92	964.15	17.68
50	85	25	0.45	4.50	9.6	1155.87	19.68
50	85	25	0.5	5.44	9.92	1030.83	22.12
50	85	25	0.55	4.12	13.72	1828.49	23.72
50	85	25	0.6	5.45	11.72	1420.99	25.56
50	85	25	0.65	6.33	13.56	2811.34	30.24
50	85	25	0.7	6.97	11.88	—	26.08
50	85	25	0.75	8.03	12.16	—	25.96
50	85	25	0.8	11.52	12.04	—	27.76

Table 5. Solution times (in seconds) and objective values for problem (8) on Erdos-Renyi graphs with various values of edge failure probabilities, s = 2, and a fixed number of vertices and edges

Erdos-Renyi Graphs				s = 2	
$\|V\|$	$\|E\|$	|N|	Probability of edge failure	Time(s)	Objective
75	295	25	0.9	10.10	12.16
		75	0.9	20.92	12.16
75	295	25	0.8	13.51	13.20
		75	0.8	31.38	12.91
75	295	25	0.7	16.56	14.48
		75	0.7	44.71	13.97
75	295	25	0.6	13.00	15.84
		75	0.6	68.73	14.71
75	295	25	0.5	14.27	17.24
		75	0.5	84.60	15.81
75	295	25	0.4	24.11	18.16
		75	0.4	117.02	16.79
75	295	25	0.3	30.98	19.48
		75	0.3	151.00	19.53
75	295	25	0.2	48.89	20.60
		75	0.2	166.86	22.17
75	295	25	0.1	80.71	21.92
		75	0.1	195.90	21.20

Table 6. Parameter settings and computational statistics for problem (9) with $q = 1$

$\|V\|$	d	$\|K\|$	q	θ	c_i	t_k	# instance solved	Avg time (s)	Avg gap (%)
10	0.2	10	1	0.3	5	3	100	0.02	0%
10	0.3	10	1	0.3	5	3	100	0.07	0%
10	0.4	10	1	0.3	5	3	100	0.46	0%
50	0.04	30	1	0.4	7	4	100	0.61	0%
50	0.05	30	1	0.4	7	4	100	2.54	0%
50	0.08	30	1	0.4	7	4	20	858.33	5.80%
100	0.03	60	1	0.4	8	5	100	12.56	0%
100	0.05	60	1	0.4	8	5	57	1994.77	4.99%

Table 7. Parameter settings and computational statistics for problem (10) with $q = 1$

$\|V\|$	d	$\|K\|$	θ	c_i	t_k	# solved	TTO	# TL	# TL, inf.	Avg gap (%)
10	0.2	10	0.5	4	4	10	1.22	0	0	-
10	0.3	10	0.5	4	4	10	1.18	0	0	-
10	0.5	10	0.5	4	4	10	3.22	0	0	-
15	0.2	15	0.5	4	4	0	–	10	0	9.40%
15	0.3	15	0.5	4	4	0	–	10	0	13.47%
15	0.5	15	0.5	4	4	0	–	10	0	21.96%
20	0.2	20	0.5	4	4	0	–	10	4	37.49%
20	0.3	20	0.5	4	4	0	–	10	0	10.05%
20	0.5	20	0.5	4	4	0	–	10	8	71.96%

Table 8. Mean and standard deviation of positioning error results

	Actual	Estimated	
Inquiry SAR Patch	Coordinates [*deg*]	Coordinates [*deg*]	Error [m]
Fig.2(a)	38.0625, -122.2733	38.0625, -122.2734	5.7288
Fig.2(b)	37.9836, -122.3599	37.9836, -122.3600	5.7347
Fig.2(c)	61.0926, -164.1878	61.0926, -164.1879	4.2529
Fig.2(d)	61.0808, -164.1208	61.0808, -164.1208	5.0970

DISTRIBUTION LIST

DTIC/OCP 8725 John J. Kingman Rd, Suite 0944 Ft Belvoir, VA 22060-6218	1 cy
AFRL/RVIL Kirtland AFB, NM 87117-5776	1 cy
Official Record Copy AFRL/ RVS/Thomas A. Lovell	1 cy

(This Page Intentionally Left Blank)