12TH ICCRTS
"Adapting C2 to the 21st Century"

Locating Optimal Destabilization Strategies

Track 2, 3 or 5

Il-Chul Moon, Student, and Kathleen M. Carley

Point of Contact: Il-Chul Moon
Center for Computational Analysis of Social and Organizational Systems
5000 Forbes Avenue, Pittsburgh, PA, 15213
Tel: 1-412-268-3163
imoon@andrew.cmu.edu and carley@cs.cmu.edu

This paper is under the A2C2 program with Gerald Malecki and Rebecca Goolby

Locating Optimal Destabilization Strategies

Il-Chul Moon and Kathleen M. Carley
School of Computer Science, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA, USA, 15213
imoon@andrew.cmu.edu, carley@cs.cmu.edu

***Abstract*— Network destabilization is a critical tactic for disrupting organizations such as terrorist networks or organized crime networks. For any network, there are many possible tactics that could be used to destabilize it. For example, a set of nodes (people) could be removed or added. So could the links among the nodes. Even if we limit ourselves to node removal, there are issues which nodes to remove and when. Thus, an automated framework for the generation of the destabilization scenarios is useful, for example, in what-if scenario analysis and vulnerability analysis. We created 1024 randomly generated strategies and 1024 learning algorithm-based strategies for the destabilization of a sample terrorist network. The learning algorithm based strategies were more effective in reducing the performance of the network. Moreover, we analyzed the preference of the learning algorithm and discovered efficient tactics such as undermining the network first and isolating the knowledge sources of a network later. We believe that the presented method can be applied to other disciplines requiring a simulation and a what-if scenario generation involving network destabilization.**

I. Introduction

Network destabilization is an important issue in counter-terrorism [9, 11], network centric warfare [10], computer network security [1], etc. Also, finding efficient tactics to attack a network is critical in network destabilization. For example, if one can destabilize a computer network successfully with limited hacking attempts, the hacker will have a greater chance to hide his traces in the network. In another example, if an agency can disrupt a terrorist network with less frequent strategic interventions, the agency will achieve its goal more efficiently.

Not only the number of interventions, but also the efficiency of an intervention is critical. Commanders trying to destabilize a terrorist network often need an answer about whether or not capturing a terrorist will disrupt the performance and the plan of the network. Eliminating the threat directly causing the crisis may be able to mollify the situation, but it may emerge a more dispersed, noisier and more unpredictable terrorist network after removing a certain agent in the network. In other words, the isolation of agents will cause diverse effects on the network. Some of the effects might be preferable while others may cause unexpected and undesirable backlash.

These difficulties with the attempts at network destabilization can be partially resolved by adopting several analysis methods [6]. Social network analysis has been used to identify key persons for a network. Its methods and measures capture the various aspects of a network and helps analysts understand the network and its characteristics. Multi-agent simulation provides an ability to replicate the emergent behavior of agents, and the behavior often gives insights into what will happen with a scenario or a hypothesis. However, these solutions are not complete enough to devise destabilization strategies and test them to a certain extent if they are used separately. Furthermore, given the vast size of the possible strategy choice space, there should be a mechanism to converge efficient strategies.

In this paper, we integrated the methods from three different domains: social network analysis, multi-agent simulation and machine learning. We develop destabilization strategies with a machine learning algorithm using the analysis measures of social network analysis, and we test the efficiency of the strategies with a multi-agent simulation. With the proposed method, we expect to see that the learning algorithm based strategies are more efficient in destabilizing a network. Furthermore, the whole procedure requires no human intervention except the input of the target network, so we can apply the method to any network structured organizations.

II. Previous research

The concept and the importance of network destabilization are well documented in *Networks and Netwars* [2]. Current terrorist or criminal groups are actually leaderless although the members of the groups are able to assemble rapidly and operate as a well-functioning organization. Arquilla and Ronfeldt examined many different netwars ranging from social activist groups to violent terrorist groups. They found five major aspects of these groups, technological, social, narrative, organizational and doctrinal, of these networked organizations and netwars against them in the analysis. Particularly, the discussion of the importance of social basis for cooperation among the network members is interesting because our analysis fundamentally depends on the importance of the social structures. They argued that a network's effectiveness increases when it has built mutual trust and identity based on strong social ties. In other words, weakening the social ties is the start to undermine the terrorist or criminal networks. In addition, they claim the importance of network structure recognition and the methods for the network analysis, which brings in social network analysis and other cutting edge methods. We think that

the manuscript described the demand for network destabilization and the possible research efforts to analyze this new area.

The paper that directly motivated this research is *Destabilizing Networks* [4]. It discusses the capability of current tools and the difficulties of the destabilization problem. The current tools developed in social network analysis are introduced. The tools consisted of various network measures, such as centralities, and statistical analysis programs, like UCINET, for the measures. Furthermore, it introduces methods for comparing network structures and finding patterns from the changes. This pattern recognition technique is helpful for identifying missing links and nodes, network build mechanism, etc, which are fundamental in the vulnerability analysis of network structures. Additionally, this paper proposes a possible support from multi-agent simulation. It presents a set of examples, like analyzing network destabilization, information diffusion, bioterrorism attack, assisted by multi-agent simulation. They also mention the difficulties in the research of destabilization. The difficulties mostly live in the nature of networked organization, distributed resource and knowledge and ever-changing network shapes and internal dynamic. Carley et al claims that these difficulties can be overcome by sharpening the current tools. For example, expanding social network analysis to dynamic network analysis including not only agents but also whole and changing ingredients of a networked organization is an important to capture the changes of a network, and multi-agent communication simulation supported by network analysis will capture the evolution of the dynamically changing networks. This paper partially implemented the suggested elements of these possible improvements and the current cutting edges in the research method.

There has been a number of practical research projects addressing network destabilization. Among them, the project similar to this paper is *Netwatch* [16]. In Netwatch, there are two teams, red team and blue team, and they try to execute assigned tasks by gathering necessary knowledge and resource and to come up with a network structure emerged from a set of signal intelligence respectively. Also, the secondary object of the blue team is destabilizing or decreasing the performance of the red team, which perfectly corresponds to our research goal. However, the destabilization tactics of Netwatch is too simple to represent the real world. The author assumes four tactics: no attack, isolating the member of the red team, isolating the member with the highest degree centrality and isolating the member with the highest cognitive load. In this tactics, there are no multiple isolations and no dynamic changes in the isolations though we often isolate a set of agents and the removal often causes the structural changes of the networks. Therefore, we inherit the fundamental methods of the experiment, such as use of social network analysis, multi-agent simulation, concept of isolation, etc. Nevertheless, we expand the tactic determination by including a machine learning aspect in the strategy generation and increasing the number of isolated agents in a network sequentially.

Truly, the isolation of multiple agents, not a single agent, will change the fundamental of the network destabilization tactic because of the lock-in situations of action. First, network healing effect prohibits a static tactic of isolation. According to Carley [6], when a network gets damage by missing links or nodes, its nodes interact with each other and create alternative ways of communication and cooperation. Thus, single isolation does impact the performance of the network, but not significant enough to dismantle the network. The multiple isolations should take the network healing into account to maximize their effect. Second, a static tactic may fall into the escalation of commitment to a course of action [3]. If we stick to a single plan such as isolating agents with exclusive knowledge, we may only get a growing number of agents to isolate as the agents interact more and diffuse their information throughout the network. Therefore, we have to balance the various tactics like destabilizing the interaction of agents and cutting-off the source of information, etc. The difference between the single agent isolation and the multiple isolations bring in more complex isolation case generator and evaluation criteria, and we implement that in this research.

III. METHOD

We devised a set of approaches to achieve our research objective, creating a sequence of node isolations to destabilize a network. First, we use Near-Term Analysis and Dynet as a tool to test our isolations. Second, we setup three evaluation criteria to assess the result of the isolations. These evaluation criteria show whether and how much a network is destabilized. Finally, we introduce a set of procedures to create an isolation sequence that results in the sub-optimal destabilization of a network.

A. Near-Term Analysis and Dynet

In this paper, we hypothesize that the removal of nodes from a network can induce a destabilization of a network function. Particularly, the target network will be a social network of agents, either humans or computer nodes, which can diffuse knowledge pieces to the other agents in the network. Also, destabilization means the state of a network that cannot diffuse knowledge anymore or can diffuse it in low efficiency. Therefore, testing the efficiency of knowledge diffusion should be done for each isolation sequence. To perform such a test, there have been two distinct methods: social experiments with human participants and multi-agent simulations. We chose to use the multi-agent simulation approach because of its low cost, short experiment time and easiness of performance recording.

For that reason, we used Near-Term Analysis and Dynet in our research. Dynet [7] is a multi-agent network simulation that imitates the knowledge diffusion among networked agents. It assumes that agents in a network have assigned tasks and they try to interact with other agents to gather all the necessary knowledge pieces to perform their tasks. Because it is a discrete event simulation, we can measure agents' degree of knowledge diffusion at a certain simulated time-point. Furthermore, Dynet supports a function that can isolate a set of agents at a designated time, which fits well in our research scenario.

Near-Term Analysis [12] is a wrapping function, i.e. a GUI front-end and callable from ORA [13], for Dynet. Though Dynet provides most of the needed functions for our analysis, we still have to calculate the aggregated output performance across agents in a network and to control the parameters, such as the replication number of a simulation, the input of isolation information to Dynet, etc. Therefore, we use the combination of Near-Term Analysis and Dynet to test our isolation scenarios and retrieve the consequence of the scenarios from simulations. The details about Near-Term Analysis and Dynet can be found in [12].

B. *Evaluation of an isolation sequence*

As we discussed in the previous section, we defined the destabilization of a network as the state of network that cannot diffuse knowledge or can do so with very low efficiency. Also, our test-bed provides a performance measure, knowledge diffusion [5, 12], over the course of a simulation. Thus, our evaluation of an isolation sequence depends on the movement of the knowledge diffusion. Therefore, firstly, we can use knowledge diffusion rates at the end time of simulations as a score for the isolation strategies. This is a number showing how much a network can diffuse knowledge across the structure. For the interpretation of the score, the lower rate of a strategy demonstrates the superiority of the strategy compared to others with higher rates.

Also, we setup states of a network based on its information diffusion efficiency and used as another evaluation score for isolation sequences. Specifically, the following shows three possible results, also described in Figure 1, after a single isolation happened.

1) Suppression: Suppression in this context stands for the decrement of the growth rate in the diffusion measure compared to the non-isolation case. In other words, the knowledge diffusion is increasing compared to the scenario's own previous rate, but the increment is less than that of the non-isolation case.

2) Damage: Damage implies that the knowledge diffusion is decreased compared to the rate of a previous time-point.

TABLE I
THE USED NETWORK MEASURES FOR THE LEARNING ALGORITHM, THESE MEASURES AND THE ISOLATION TIMING ARE THE INPUT FEATURES FOR THE TRAINING

	Used measures
Network measure (27 measures)	knowledge Task Completion, knowledge Under Supply, overall Task Completion, performance As Accuracy, average Distance, average Speed, betweenness Centralization, closeness Centralization, clustering Coefficient, communicative Need, connectedness, density, diameter, efficiency, fragmentation, global Efficiency, hierarchy, in Degree Centralization, lateral Edge Count, minimum Speed, network Levels, out Degree Centralization, reciprocal Edge Count, sequential Edge Count, span Of Control, strong Component Count, weak Component Count
Node measure (11 measures)	Cognitive demand, total degree centrality, clique count, row degree centrality, eigen vector centrality, betweenness centrality, high betweenness and low degree, task exclusivity, knowledge exclusivity, resource exclusivity, workload

However, the change in diffusion may not be sustained. If there is no isolation, the knowledge diffusion always increases or stays at the same level. Thus, the decrement of the diffusion rate always implies that Suppression is happening. Even though there is Suppression, it does not necessarily cause the decrement of the diffusion because a network can still develop the diffusion rate limitedly. Therefore, this is worse than Suppression.

3) Break: Break means that the amount of knowledge diffusion has dropped and has remained low for a sustained number of time periods. Even the non-isolation case often stops the increase of the knowledge diffusion after a certain amount of time-points, but there is no damage in such a case. On the other hand, there are cases that show large damage in the rate and the rate never grows. We regard this particular case as Break state of a network.

Though we defined the states after a single isolation, we will add the happenings of each type of the above events and use the sums to evaluate isolation sequences. If the happenings of the events are frequent in a simulation of a sequence, it shows the

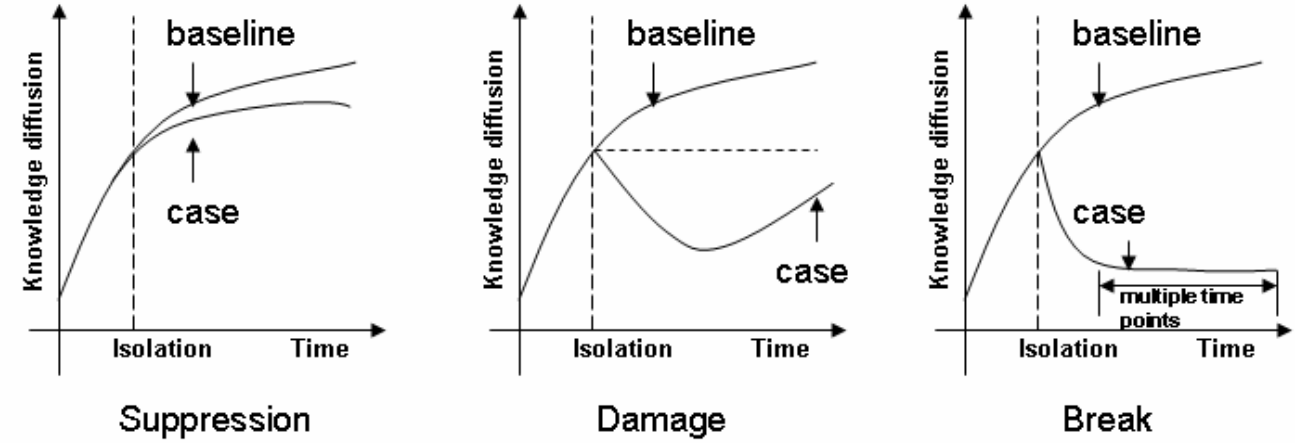

Fig. 1. three events of destabilization, event occurrences during a simulation are counted for each case and used for the evaluation of strategies. Not only the event counts, but also the end time gap between the baseline and a strategy is also considered.

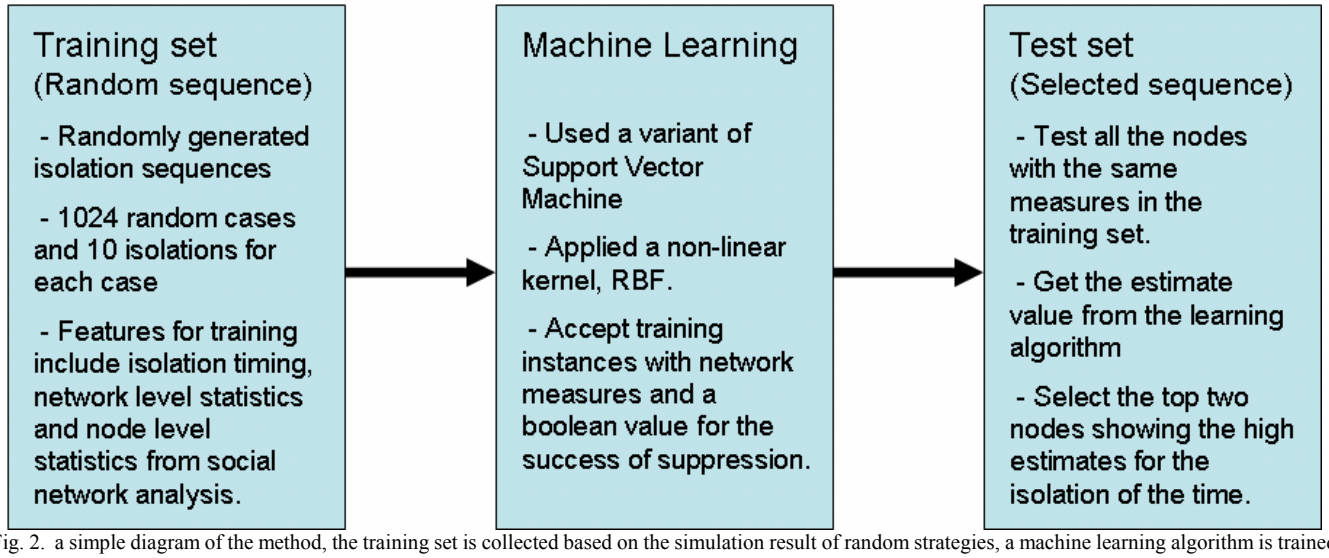

Fig. 2. a simple diagram of the method, the training set is collected based on the simulation result of random strategies, a machine learning algorithm is trained with the training set, and the selected strategies are composed based on the learning algorithm

sequence is inducing many destabilization events of the network.

C. *Generation of an isolation sequence*

The test network, which we will introduce in the next section, has only 16 agents. For this network, we will isolate ten agents out of the sixteen agents from simulation time 2 to 20. Each isolation event will have two gaps of time-points between the previous and the next isolations. However, the number of possible isolation sequences, whose size is ten, is *P(16,10)* (=5765760). Because there are too many possible isolation sequences, we will use a machine learning approach to create a sub-optimal sequence. The diagram in Figure 2 explains the procedures for the creation.

First, we created 1024 sample isolation sequences by choosing 10 agents to isolate randomly. These sample isolation sequences were tested by Dynet, a multi-agent simulation, and produced 10*1024 isolation results. Then, we divided the results into two classes, the results displaying Suppression state and the results showing no Suppression state, which includes Damage or Break state. After this sub-procedure, we obtained 10*1024 training cases with '1' or '0' output state, in which '1' stands for Suppression and vice versa.

Second, we trained a SVM Regression algorithm developed by Smola and Scholkopf [14,15]. We fed the learning algorithm input features such as target node measures, network measures before the isolation and isolation timing (listed in Table 1). Further information, i.e. formula, description and interpretation, of the measures can be found in [13], and the measure calculation is done by *Organization Risk Analyzer* developed by them. Of course, we marked the input features with an output state that is described above. Afterward, we utilized the marked training set to optimize the SVM Regression algorithm. Additionally, we used RBF kernel for the SVM with a parameter, 0.01.

Third, we started creating the isolation sequence by utilizing

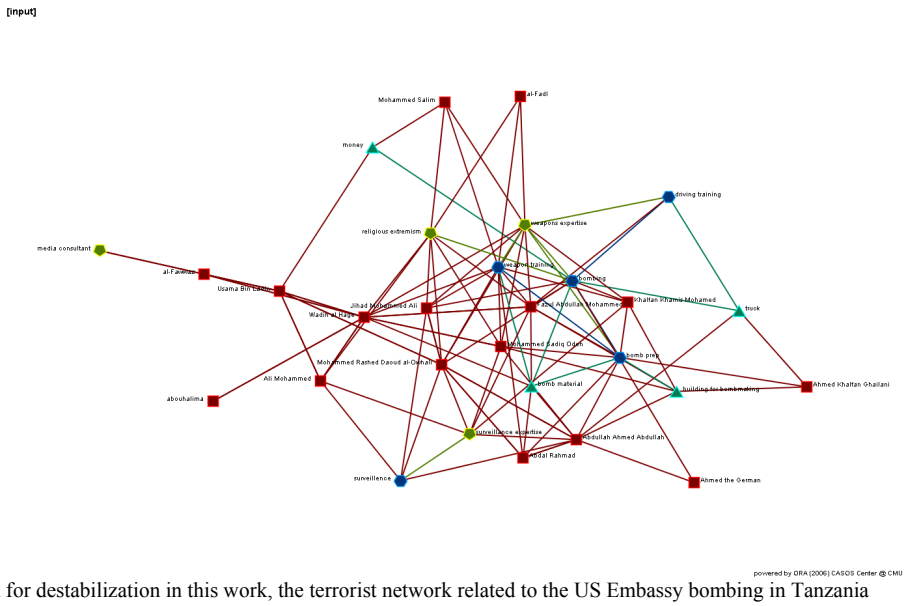

Fig. 3. the target network for destabilization in this work, the terrorist network related to the US Embassy bombing in Tanzania

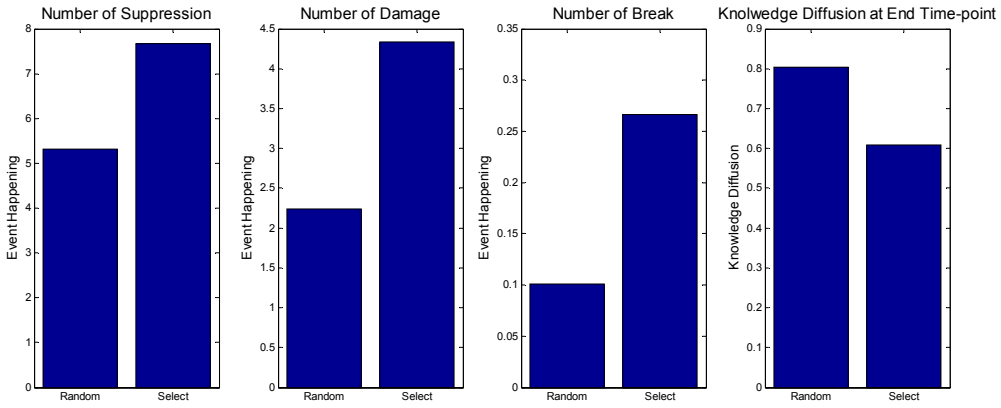

Fig. 4. the simulation result of the random strategy set and the selected set. The selected set was able to induce more destabilization events and reduce the end time knowledge diffusion.

the optimized machine learning algorithm. Because the learning algorithm is a non-linear regression, it calculates the likelihood of Suppression happening for a test instance consisting of node measures, network measures and isolation timing. Therefore, we populated test instances by using every available agent in the network at a certain time-point. With the populated test instances, the regression algorithm provided estimates for the likelihoods for the instances and we chose the two instances with the two highest likelihoods. The chosen two instances indicate two agents to isolate, and we created two possible choices at the time-point. Surely, the chosen agents will be removed from a network for the next iteration. By repeating this iteration ten times, we got 2^{10} (=1024) sequences generated by the machine learning algorithm.

Finally, we had 1024 randomly generated scenarios and 1024 machine learning produced sequences. We calculated their end-time knowledge diffusion rate, the number of Suppression events, Damage events and Break events. Also, we searched for the best isolation sequence for both generation schemes and drew their knowledge diffusion changes over the course of the simulation periods.

IV. Result

The dataset used to test the introduced method is an organizational structure of a terrorist network from the U.S. Embassy bombing incident in Tanzania [8]. The network consists of 16 agents, 4 knowledge pieces, 4 resources and 5 tasks, and it is relatively small. We believe that the members of the network tried to complete the assigned tasks by communicating with other members to obtain their necessary knowledge and resource. With this network, we created scenarios removing agents one by one and tried to destabilize the network in terms of making the network unable to diffuse the knowledge. Figure 3 is the visualization of the network.

A. Performance of isolation strategies generated by a machine learning algorithm

First, it is important to see how much the strategies from the machine learning algorithm are better than the randomly generated strategies. As described in the method section, we created 1024 random scenarios and 1024 selected scenarios and determined three destabilization events and the end-time knowledge to consider as evaluation criteria for the strategies. The Dynet simulation results with both groups of strategies are shown in Figure 4. The numbers of three destabilization events, suppression, damage and break, of the selected scenarios exceeds those of the random scenarios, which demonstrates that the selected scenarios created by the learning algorithm caused the damaging events more often. Furthermore, the average knowledge diffusion of the selected strategies is less than the average of the random scenarios. This implies that the selected strategies were successful in terms of reducing the knowledge diffusion rate by effectively isolating agents from the network.

Besides of the overall results, Figure 5 shows the diffusion changes over the simulated time periods. First, the upper graph of Figure 5 represents the average knowledge diffusion for each simulated time-point. In the graph, we noticed that the non-isolation case does better than the cases with isolations and the difference between the non-isolation case and the random isolation cases is much smaller than that between the non-isolation case and the selected cases. This result means that the random isolation would not be able to impact the diffusion rate significantly compared to the learning algorithm-based isolations.

Moreover, the lower graph of Figure 5 shows the best destabilization strategies of the two groups and the non-isolation case. The best selected strategy completely broke the diffusion among the agents, so the network could not overcome the isolations after the time-point 20 when the tenth and last isolation occurred. On the other hand, the best destabilization of the random strategy did reduce the knowledge diffusion rate, but it did not break the diffusion completely. From the movement of the best destabilization of the selected case, we see that the strategy prevented further spreading of the knowledge set in the network first and isolated the agents who already possessed the diffused knowledge.

B. Target selection of the machine learning approach

The above results demonstrate the capability of the machine

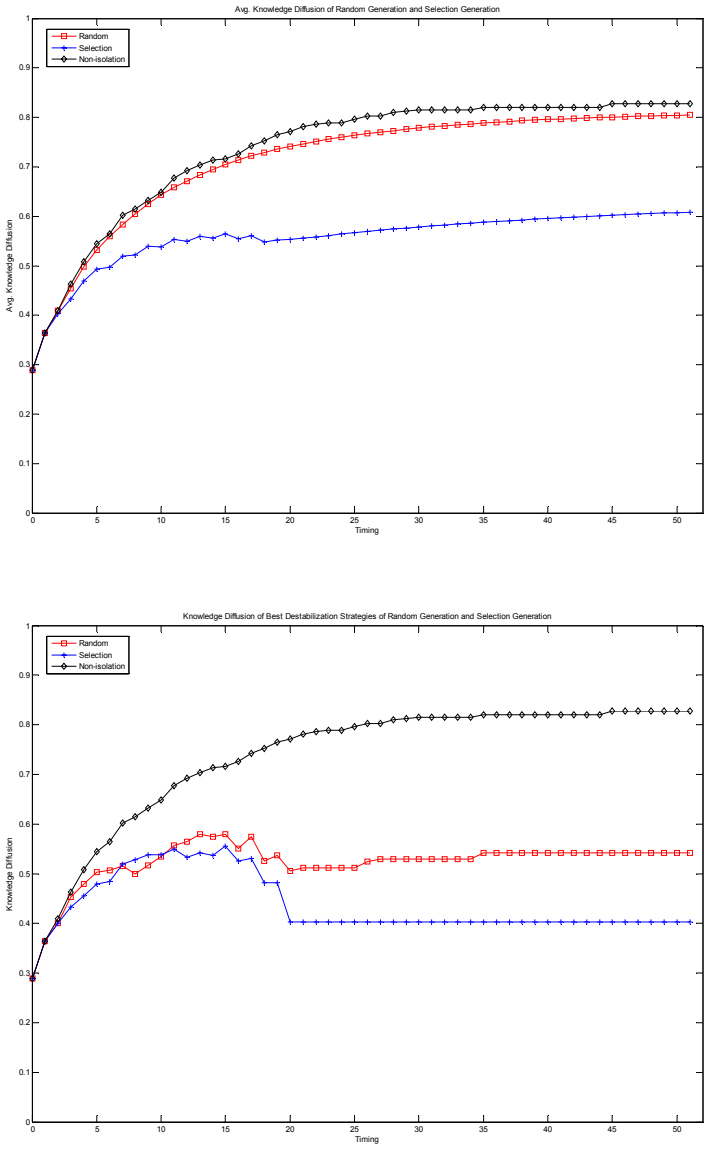

Fig. 5. (Upper) Three lines, baseline, the average of 1024 random sequences and the average of 1024 selected strategies, of average knowledge diffusion over time (Lower) same three lines but showing the best destabilization of the random and the selected sets.

learning approach for an effective isolation strategy generation. Then, the following question would be how the learning algorithm decides the target and which agent the learning algorithm isolates. The learning algorithm takes inputs from the network status and the node position when the isolation occurs. The status and the position are represented by a set of network measures calculated at the network level and the node level. Also, the position of a specific node is important in deciding who to isolate because the overall shape of the network, the topology, is the same for all of the nodes. Node position is the feature distinguishing each agent. Figure 6 is a group of graphs showing the changes of network position measures of 10 consecutive isolations. In other words, each measure represents an aspect of the node's position Conspicuously, the measures of the learning algorithm show an average tendency over the course of the simulations. Whereas the random method chooses the agents without any tendencies, the learning algorithm selects different agents at different times. At the beginning of the simulation, the learning algorithm isolates the agents with high total degree centrality, which means the isolation removes comparatively large numbers of links from the network. This trend lasts for the first three isolations. After that, the learning algorithm starts isolating connecting nodes that have high betweenness centrality and low degree centrality. As their measure name shows, the connecting nodes are the agents positioning themselves at the bridging points among groups in the network with a small number of links. As a result of isolating the two types of agents sequentially, we get the optimal destabilization result. Also, it should be noted that the agents with exclusive knowledge pieces are isolated in the

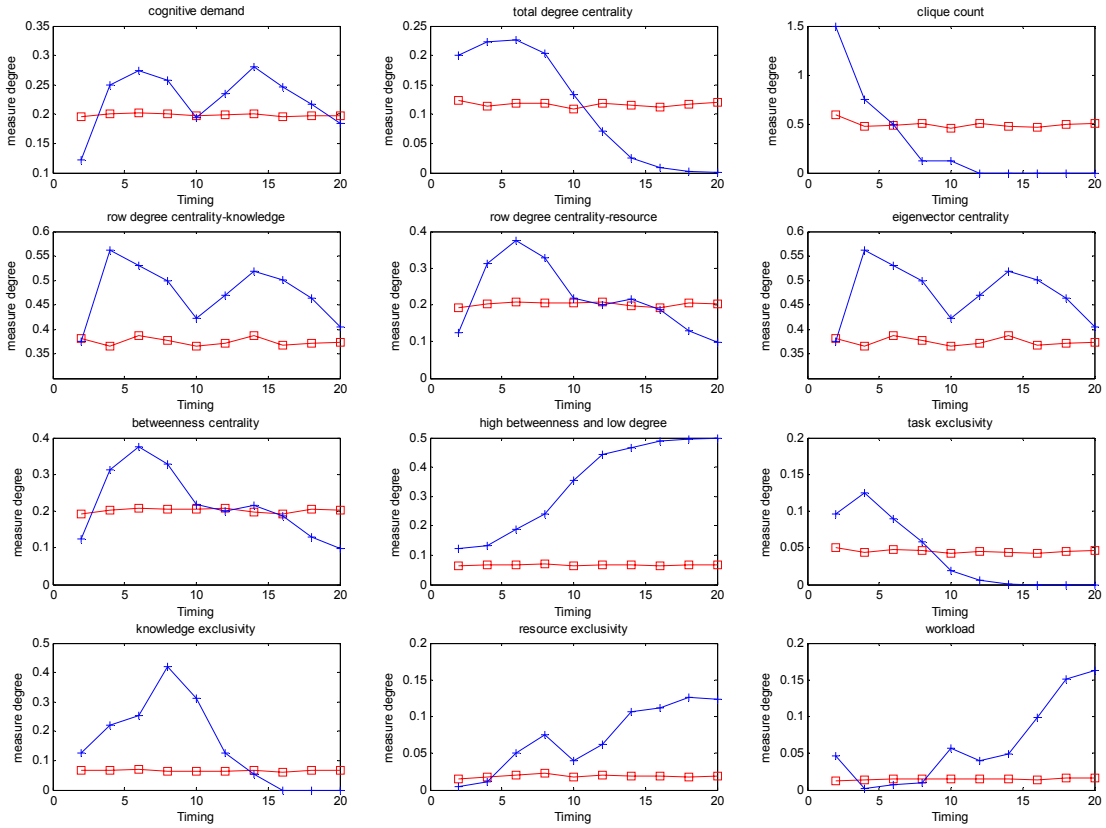

Fig. 6. the average measure changes of the selected nodes over time. a measure level of a time-point represents the preference of the learning algorithm when it tries to isolate a node at the time.

middle of the sequence. This means that the isolation of the knowledge source may not be the first task for preventing the spread of the knowledge. Rather than isolating the knowledge source agents, the learning algorithm prefers to disconnect the network first and deal with the knowledge sources after the first wave of isolations.

V. Conclusion

Network destabilization is one of the most important issues in counter terrorism, organized crime network, etc. If we are able to disrupt the information flow among the terrorists in a network, we can diminish their performance. In this paper, we assumed that an isolation of an agent is one of the ways to disrupt the network, and we researched how to devise an automated method for generating optimal destabilization strategies based on the given network structure. We utilized mainly three theories from different disciplines. First, we used a multi-agent simulation, Dynet, to replicate the communication among the agents in the network. Second, we utilized a machine learning algorithm, Support Vector Machine, to learn the most devastating isolation case and to generate the chain of the isolations. Finally, we used social network analysis measures, such as centrality measures, to numerate the network status and the agent positions on the network and to feed the information to the learning algorithm.

We setup 1024 randomly created scenarios and 1024 learning algorithm-based strategies. Also, we developed three destabilization events and one numeric score: Suppression, Damage, Break, and the end-time knowledge diffusion rate. Then, we tested the strategies with the simulation and retrieved the evaluation result. The result presents that the learning based strategies exceed random strategies in all of the four evaluation criteria. The trained machine learning algorithm was capable of creating scenarios that destabilize a network better by inducing the destabilization events more often and reducing the end time knowledge diffusion rate compared to the random scenarios. Furthermore, we analyze the isolation target choice tendencies of the learning algorithm. The learning algorithm tends to isolate high total degree centrality nodes and then connecting nodes. The isolation of knowledge sources happened after three or four isolations. In other words, the learning algorithm removed most of the links in the network first and the knowledge later from the network. By doing so, the learning algorithm minimized the spread of the knowledge pieces across the network. At the same time, the learning algorithm was capable of isolating a limited number of nodes with already spread knowledge pieces after making the network unable to diffuse the knowledge.

This research method does not have to be limited to a destabilization method of a terrorist network. It can be applied to the destabilization of a network centric warfare system, a government structure, a computer network, etc. Because there is no human intervention in creating the strategies, the worst case scenario generation can be done only with an input of an organization structure. Then, the learning algorithm will search the domain specific parameters by repeating simulations. On the other hand, the usage of the simulation should be validated in the used domains to gain credibility. Dynet has been validated and utilized to test the destabilization of terrorist networks, military command and control, corporate management, etc.

This multi-disciplinary approach can be improved by enhancing each element in the method. The learning algorithm is not examined for its own test set accuracy, and very little work is done to enhance its training accuracy. The possible parameters and the applicable kernel of the learning algorithm are vast, which give us room for improvement. In addition, we should develop new social network measures that can capture the salient features of network changes. The robustness of a network measure can contribute to the stable input for the learning algorithm when the network is dynamically changing. Lastly, the multi-agent simulation, Dynet, should be validated in many different disciplines and adopt more realistic experiment functions to expand the usage of this approach. Any of these improvements increase the usability and the accuracy of the proposed method.

This paper proposed an automated method of network destabilization strategies and tested the generated strategies with a multi agent simulation. The test result shows that the generated strategies are better than randomly generated strategies in terms of prohibiting information flow in a network. This method can be applied to any domain involving an organization and information diffusion.

ACKNOWLEDGMENT

This work was supported in part by the Office of Naval Research (ONR N0001140210973-NAVY, N000140610921 and N00014-06-0104), the National Science Foundation (SES-0452487), the Army Research Lab, and the AirForce Office of Sponsored Research (MURI: Cultural Modeling of the Adversary, 600322) and the Department of Defense for research in the area of dynamic network analysis. Additional support was provided by CASOS - the center for Computational Analysis of Social and Organizational Systems at Carnegie Mellon University. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the Office of Naval Research, the National Science Foundation, the Army Research Lab or the U.S. government.

REFERENCES

[1] R. Albert, H. Jeong and A.-L. Barabasi (2000), Error and attack tolerance of complex networks, Nature, Vol 406, pp 378-382

[2] J. Arquilla and D. Ronfeldt (editors) (2001), Networks and Netwars: The Future of Terror, Crime, and Militancy. Santa Monica, Calif.: RAND, MR-1382-OSD. www.rand.org/ publications/ MR/MR1382/

[3] J. Brockner (1992), The escalation of commitment to a failing course of action: Toward theoretical progress, Academy of Management Review, Vol 17, pp 39-61.

[4] K. M. Carley, J. S. Lee and D. Krackhardt (2001) Destabilizing Networks. Connections, Vol 24(3), pp 31-44.

[5] K. M. Carley and C. Schreiber (2002), Information Technology and Knowledge Distribution in C3I teams, Proceedings of the 2002 Command and Control Research and Technology Symposium, Naval Postgraduate School, Monterey, CA:

[6] K. M. Carley (2003), Dynamic Network Analysis, Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers, Eds. Ronald Breiger, Kathleen Carley, and Philippa Pattison, Committee on Human Factors, National Research Council, National Research Council, pp 133-145

[7] K. M. Carley (2004), Estimating Vulnerabilities in Large Covert Networks Using Multi-Level Data, In Proceedings of the 2004 International Symposium on Command and Control Research and Technology. Conference held in June, San Diego, CA., Evidence Based Research, Presented during Track 1, Electronic Publication, Vienna, VA.

[8] K. M. Carley and K. Y. Natalia (2004), A Network Optimization Approach for Improving Organizational Design, Carnegie Mellon University, School of Computer Science, Institute for Software Research International, Technical Report CMU-ISRI-04-102.

[9] K. M. Carley (2006), Destabilization of covert networks, Computational & Mathematical Organization Theory, Vol 12. Num 1., pp 51-66

[10] A. H. Dekker and B. D. Colbert (2004), Network robustness and graph topology, Proceedings of the 27th Australasian conference on Computer science, Vol 26, pp 359-368

[11] J. S. McIllwain (1999), Organized crime: A social network approach, Crime, Law and Social Change, Vol 32. Num 4., pp 301-323

[12] I. C. Moon and K. M. Carley (2006), Estimating the near-term changes of an organization with simulations, AAAI Fall Symposium, Arlington, VA

[13] J. Reminga and K. M. Carley (2004), ORA:Organization Risk Analyzer, Tech Report, CMU-ISRI-04-106, CASOS. Carnegie Mellon University. Pittsburgh PA, http://www.casos.cs.cmu.edu/projects/ora/index.html

[14] S. K. Shevade, S. S. Keerthi, C. Bhattacharyya and K. R. K. Murthy (1999), Improvements to SMO Algorithm for SVM Regression. Technical Report CD-99-16, Control Division Dept of Mechanical and Production Engineering, National University of Singapore.

[15] A. J. Smola and B. Scholkopf (1998), A Tutorial on Support Vector Regression, NeuroCOLT2 Technical Report Series - NC2-TR-1998-030.

[16] M. Tsvetovat (2005), Social structure simulation and inference using artificial intelligence techniques, Ph. D. Thesis, Carnegie Mellon University, CMU-ISRI-05-115