Developing Proactive Defenses for Computer Networks with Coevolutionary Genetic Algorithms

Anthony Erb Lugo
MIT, CSAIL
32 Vassar St
Cambridge, Massachusetts 02142
aerblugo@mit.edu

Dennis Garcia
MIT, CSAIL
32 Vassar St
Cambridge, Massachusetts 02142
dagarcia@mit.edu

Erik Hemberg
MIT, CSAIL
32 Vassar St
Cambridge, Massachusetts 02142
hembergerik@csail.mit.edu

Una-May O'Reilly
MIT, CSAIL
32 Vassar St
Cambridge, Massachusetts 02142
unamay@csail.mit.edu

ABSTRACT

Our cybersecurity tool, RIVALS, develops adaptive network defense strategies by modeling adversarial network attack and defense behavior in peer-to-peer networks via coevolutionary algorithms. Currently, RIVALS DOS attacks are modestly modeled by the selection of a node that is completely disabled for a resource-limited duration. Defenders have three different network routing protocols. Attack or mission completion and resource cost metrics serve as attacker and defender objectives. This work also includes a description of RIVALS' suite of coevolutionary algorithms that explore archiving as a means of maintaining progressive exploration and support the evaluation of different solution concepts. To compare and contrast the effectiveness of each algorithm, we execute simulations on 3 different network topologies. Our experiments show that it is possible to forgo the assurance of monotonically increasing results and still retain high quality results.

CCS CONCEPTS

•**Computer systems organization** → **Embedded systems;** *Redundancy;* Robotics; •**Networks** → Network reliability;

KEYWORDS

cybersecurity, coevolution, network, evolutionary algorithms

ACM Reference format:
Anthony Erb Lugo, Dennis Garcia, Erik Hemberg, and Una-May O'Reilly. 2017. Developing Proactive Defenses for Computer Networks with Coevolutionary Genetic Algorithms. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 2 pages.
DOI: http://dx.doi.org/10.1145/3067695.3089234

GECCO '17 Companion, Berlin, Germany
 978-1-4503-4939-0/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3067695.3089234

1 INTRODUCTION

Cyber attacks have increased in frequency and severity and have been the cause of numerous disruptions in both industry and politics. With more and more critical information moving through networks, defenses must be in place to help keep these networks secure. Moreover, when an attacker is deterred by a specific defense, the attacker usually changes strategies. Thus, defenders have to adjust to these new attacks and a perpetual repetetive adversarial process escalates.

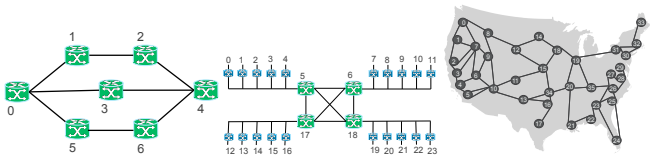
We introduce a new cybersecurity project, RIVALS. RIVALS uses coevolutionary algorithms to determine the best defense for a network amidst constantly changing cyber attacks. RIVALS focuses on a peer-to-peer network as a robust and resilient means of securing mission reliability against distributed denial of service attacks.

RIVALS will eventually include a peer-to-peer network simulator of an extended version of the Chord [3] protocol. Now, we model simple attacks and defenses on the network. We measure the performance of attackers and defenders through the concept of a mission. A mission is a set of tasks to be completed. These tasks rely on the network's health for their success. An attacker's goal is to degrade the network so tasks, and the mission, fail. Meanwhile, a defender's goal is to ensure mission success. To model the adaptive behavior of adversaries in network security, we consider our attacking and defending algorithms as populations under the direction of a coevolutionary algorithm. Over the course of many generations, this evolutionary process produces defender and attacker configurations which can then be used to determine an effective defensive protocol for a given network.

We examine the performance of different coevolutionary algorithms on RIVALS' network simulator. Additionally, we introduce rIPCA, which expands on the idea of non-domination and applies this concept.

2 METHOD

We present our methodology with respect to: coevolutionary algorithms [2]. IPCA (Incremental Pareto-Coevolution Archive) archives previous tests and only replaces them with new tests which are different and more competitive than those in the archive. This



(a) Topology 0 (b) Topology 1 (c) Topology 2

Figure 1: Topologies, used for attacks and defense..

strategy fosters monotonic evolutionary progress. Coev is a simple coevolutionary algorithm [1] which can be configured to use either the maximum expected utility solution concept or the best worst solution concept. IPCA and rIPCA both use archives and the Pareto Optimal Set solution concept. MaxSolve uses the maximum expected utility solution concept and archives. MinMax picks the best worst case.

Peer-to-peer networks have no single point of failure and thus are more inherently robust to defend against DDoS attacks. Our implementation of Chord is as a model rather than a deployed distributed network.

3 EXPERIMENTS

Our experiments seek to understand the capabilities of our coevolutionary algorithms when different solution concepts are used. They help us start to examine and interpret the resulting dynamics. We apply the suite of coevloutionary algorithms in a simple RIVALS context by setting up 3 different network topologies run with network simulation. The key parts for the simple RIVALS network simulator are:

Network Topology We start with a simple topology (Figure 1a) to exhaustively search through all attack and defense scenarios and then scale up to larger and more realistic topologies (see Figures 1b, 1c) that are too large enumerate, and require search. We assume that every edge is unit-length.

Missions: A mission is a sequence of tasks where each task specifies a start node, an end node, and a time allowed. A mission is successful if every task is completed under the time allowed and fails if any of the tasks of the mission fail. We currently limit missions to one task.

Attacker: The goal is to disrupt the network to cause mission failure, with as little effort as possible.

Defender: The goal of the defender is to ensure mission success. Currently, the defender network routing protocols are: (1) shortest path (2) flooding (3) Chord.

Fitness Functions: Attackers are rewarded for using few nodes and short duration to disrupt a mission. The attacker's fitness function is

$$f_a = \frac{1 - mission\_success}{(n\_attacks \cdot total\_duration) + n\_attacks}$$

where $mission\_success$ is describing whether the entire mission succeeded (1) or failed (0), $n\_attacks$ is the total number of nodes attacked in the network, and $total\_duration$ is the aggregated amount of time nodes were attacked.

Table 1: Coevolution results

Algorithm	Topology 0 Final Perf.	Topology 1 Final Perf.	Topology 2 Final Perf.
Simple Coev	0.227 ± 0.05	0.067 ± 0.031	0.053 ± 0.022
MinMax	0.200 ± 0.060	0.059 ± 0.013	0.057 ± 0.017
MaxSolve	0.263 ± 0.159	0.074 ± 0.026	0.081 ± 0.027
IPCA	0.333 ± 0.063	0.070 ± 0.003	0.079 ± 0.000
rIPCA	0.463 ± 0.018	0.068 ± 0.003	0.062 ± 0.000

Defenders are rewarded for completing the mission quickly and with a short amount of hops. The defender's fitness function is

$$f_d = \frac{mission\_success}{overall\_time \cdot n\_hops}$$

where $overall\_time$ is the total time a specific routing protocol took to complete the mission and n_hops is the total number of hops taken by the protocol to complete the mission.

3.1 Results

Results from our experiments are in Table 1. Additionally, we performed an exhaustive search of Topology 0. We performed this exhaustive search as a means of verifying the correctness of both the algorithms and the defense protocols in the network. Moreover, we show that this exhaustive search is possible in topologies with a small number of nodes but becomes increasingly difficult as you reach a topology as large as the one in Figure 1c. Each entry in the table represents the state of each population at the end of a run under one of the implemented coevolutionary algorithms by calculating the average fitness of the defending population under coevolution.

For Topology 0, we notice how the algorithms differ when both populations are set to evolve dynamically. The results favor IPCA and rIPCA as they seem to be better suited at finding strong population individuals. We suspect this is due to the nature of the test archives for both IPCA and rIPCA as these archives help enforce monotonic performance increases. As for the coevolutionary results, we note that both IPCA and rIPCA converged on a solution as evidenced by their low standard deviations. The remaining three algorithms have more variance.

4 CONCLUSIONS & FUTURE WORK

We introduced an end-to-end system where we have shown the ability to test the effectiveness of the different coevolutionary algorithms on simulated networks. One of our next tasks is to improve the realism of the network simulator by incrementally increasing its sophistication and complexity.

REFERENCES

[1] Erik Hemberg, Jacob Rosen, Geoff Warner, Sanith Wijesinghe, and Una-May OfiReilly. 2016. Detecting tax evasion: a co-evolutionary approach. *Artificial Intelligence and Law* 24, 2 (2016), 149–182.

[2] Elena Popovici, Anthony Bucci, R Paul Wiegand, and Edwin D De Jong. 2012. Coevolutionary principles. In *Handbook of Natural Computing*. Springer, 987–1033.

[3] Ion Stoica, Robert Morris, David Karger, M Frans Kaashoek, and Hari Balakrishnan. 2001. Chord: A scalable peer-to-peer lookup service for internet applications. *ACM SIGCOMM Computer Communication Review* 31, 4 (2001), 149–160.