Hyper-Heuristics for the Automated Design of Algorithms

Practitioners in industry and government have a huge need for high-performance algorithms customized to their very specific needs for repeated solving of instances of the same, computationally expensive, problem class. However, they often lack the expertise, or it is simply not cost-effective, to manually perform such customization.

OBJECTIVES

- Automate the design of algorithms to create highly customized solutions for repeated solving of instances of the same problem class, where high a priori computational cost can be effectively amortized
- Create tools to assist practitioners with automating the design of algorithms
- Demonstrate technical approach on real-world problems

PoC: Daniel Tauritz, Ph.D.
Associate Professor of Computer Science, Missouri S&T
Academic Director, LANL/S&T Cyber Security Sciences Institute
Contract Scientist, Sandia National Laboratories
Director, Natural Computation Laboratory
tauritzd@mst.edu, 573-341-7218

Current Projects

- Evolving SAT Solvers (funded by SNL)
- Evolving Bipartite Authentication Graph Partitioning Algorithms (funded by LANL)
- Targeting Diverse Computational Architectures (funded by SNL)
- Evolving Fast Update Algorithms for Dynamic Graph Properties (collaboration with LANL)

Technical Approach

Create a hyper-heuristic framework which:

I. Extracts Algorithmic Primitives from existing algorithms
II. Evolves novel algorithms employing Genetic Programming
III. Employs Asynchronous Parallelism to significantly speed up evolution by exploiting Solution Run-Time Heterogeneity
IV. Targets Computational Architectures employing Multi-Objective Evolutionary Algorithms

Scholarly Work

GECCO 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016

CEC 2007, COMPSAC 2014, PoS 2016 (submitted), PPSN 2016 (submitted)

Significant Technical Challenges

- How to select the most appropriate granularity for the algorithmic primitives? Automated decomposition and composition of primitives may be a solution.
- How to select the most appropriate type of Genetic Programming for a particular hyper-heuristic application is an open question in the research community.
- The subtle change in evolutionary dynamics caused by Asynchronous Parallelism are not currently well understood.
- Targeting Embedded Systems requires not yet developed, sufficiently accurate algorithm performance approximations, to facilitate evolving on High Performance Computing systems.

Coevolutionary Computational Game Theory

Game theory is a powerful tool to analyze the modeling of adversaries, such as for instance attackers and defenders in cyber security. However, classic game theory approaches are unable to scale up to analyze modern complex systems such as enterprise computer network systems and cyber-physical infrastructure systems. Co-optimization, in particular coevolution, is a promising heuristic approach to scaling game theory.

OBJECTIVES

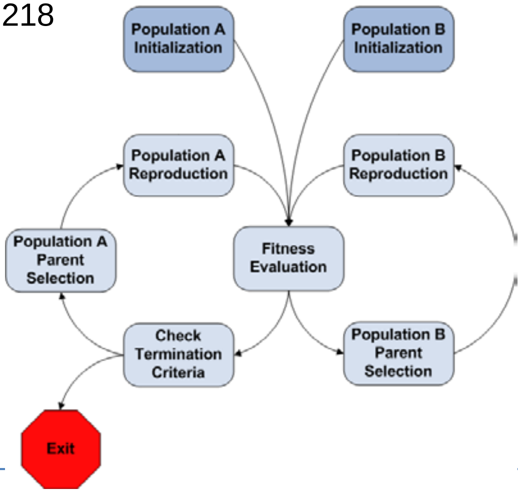
- Automate the identification of (adversarial) threats
- Automate the mitigation of identified (adversarial) threats
- Automated analysis tool development

Current Projects

- Evolving Attackers & Defenders for Enterprise Computer Networks (funded by LANL)
- Evolving Human Behavior Mimicking Agents to Enhance Computer Network Emulation Fidelity (funded by SNL)

PoC: Daniel Tauritz, Ph.D.
Associate Professor of Computer Science, Missouri S&T
Academic Director, LANL/S&T Cyber Security Sciences Institute
Contract Scientist, Sandia National Laboratories
Director, Natural Computation Laboratory
tauritzd@mst.edu, 573-341-7218

Coevolutionary co-optimization (populations A & B contain the current samples of the respective search spaces of threats and mitigations)



Technical Approach

- Create appropriate computational models of the complex systems to be analyzed
- Develop automated threat & mitigation analysis tools employing co-optimization – in particular coevolution, multi-agent systems, and machine learning

Scholarly Work

IEEE SoSE 2007, COMPSAC 2007, 2014

GECCO 2008, 2009, 2010, 2011, 2012, 2013, 2015

Significant Technical Challenges

- Application domain modeling
- High-fidelity models tend to be too slow for co-optimization
- Network simulations are too low-fidelity for analysis of enterprise computer network systems
- Emulations lack human behavioral dynamics
- Employing live malware is potentially dangerous
- How to integrate learning multi-agent systems?