# Artificial Markets for Software Innovation and Algorithmic Evaluation

Karl Lieberherr and Ahmed Abdelmeged and Bryan Chadwick

College of Computer & Information Science
Northeastern University, 360 Huntington Avenue
Boston, Massachusetts 02115 USA.
{lieber,mohsen,chadwick}@ccs.neu.edu

Date: July 10, 2009

# 1 Project Summary

Benchmarks set standards for software innovation and evaluation in many fields of computer science. Composing benchmarks is challenging: (1) Benchmarks are rigid and only represent a sample of the application behavior space, (2) Benchmark suites take several years to develop, but applications evolve at a faster rate, and (3) Benchmarks are difficult to standardize. The objective of this work is to develop a dynamic benchmark generation strategy, based on artificial markets of computational challenges, to address these benchmarking problems. A dynamic evaluation goes deeper than a static benchmark evaluation and will foster more innovation.

**Intellectual Merit:** Our two main objectives are to show that (1) artificial markets of computational challenges are a useful dynamic benchmark tool for software and (2) they are an innovation driver for finding better algorithms. We would like to help program officers, professors and managers with the evaluation of research prototypes, student programs and competing software packages, respectively. And we would like to help algorithm designers to find better algorithms through the feedback from artificial markets. When our research is complete, it will be possible to take a description of a computational domain $X$ and generate an artificial market $SCG(X)$ on the web that will be inhabited by agents offering competing software packages for domain $X$. (SCG stands for: Specker Challenge Game and is named for a 1981 Journal of the ACM publication which motivated the game.) The SCG software translates a description of $X$ into a baby agent and a trust-worthy game administrator for $SCG(X)$. Contests between the agents will be run frequently to improve the agents. Each agent gets for each contest a ranking using a group, round-robin or Swiss tournament. The agents offer and accept computational challenges consisting of a set of problems in $X$ (defined extensionally or intensionally by a predicate) and a price. The offerer will give one of those problems to an accepting agent for solution. The acceptor will recuperate the initial price based on the quality of the acceptor and offerer solutions, and the price. From a game perspective, the proposal is about a new area: meta mechanism design for algorithm design: We design a family of game mechanisms that drive the players towards better algorithms for a selected problem domain.

**Broader Impacts:** The project will advance discovery and understanding of how to solve computational problems while promoting teaching and learning. Expected outcomes will be better algorithmic techniques for a wide variety of problems and a new experiential teaching technique. When students come to the course ”Software Development for Computational Domain $X$” they are given a baby agent that can minimally survive in artificial market $SCG(X)$. They grow their baby agent into a winning agent that successfully beats the agents developed by their peers. The contests will run on the web for easy access and software, based on a state-of-the-art generic and generative programming tool called DemeterF, will be distributed widely. We have successfully used two instantiations of SCG: $SCG(Maximum\text{-}Satisfiability)$ and $SCG(Boolean\ MaximumConstraintSatisfaction)$ in undergraduate and graduate courses.

The proposed activity will benefit society through more reliable software that has been thoroughly tested and improved in artificial markets.