Steven Gustafson

General Electric Global Research Center

Steven Gustafson is a computer scientist at the General Electric Global Research Center in Niskayuna, New York. As a member of the Computational Intelligence Lab, he develops and applies advanced AI and machine learning algorithms for complex problem solving. He received his PhD in computer science from the University of Nottingham, UK, where he was a research fellow in the Automated Scheduling, Optimisation and Planning Research Group. He received his BS and MS in computer science from Kansas State University, where he was a research assistant in the Knowledge Discovery in Databases Laboratory. His PhD dissertation, an analysis of a biologically inspired search algorithm in the space of computer programs, was nominated for the British Computer Society and the Conference of Professors and Heads of Computing Distinguished Dissertation award, which recognizes the top PhD thesis in the UK computer science community. In 2005 and 2006, he coauthored papers that won the Best Paper Award at the European Conference on Genetic Programming. Outside of work, Gustafson enjoys literature and traveling with his wife and infant son. Contact him at steven.gustafson@research.ge.com.

Creative Problem Solving with Genetic Programming

Genetic programming is a heuristic search method that uses a population of variable-length computer programs and a search strategy based on biological evolution. It represents an intuitive method for automatically evolving programs. (For examples of genetic programming, see the latest editions of IEEE publications.)

Several grand challenges exist in genetic programming, one of which helps illustrate an advantage of the method. By representing solutions with programs, practitioners specify the solution primitives in familiar algorithmic building blocks. These primitives are usually similar to, or the same as, those in most high-level programming languages. Actually constructing programs from these primitives (including the program's topological structure and content) is left to the transformation operators and evolutionary process in the genetic programming search strategy. However, a very complex problem arises: how do we transform solutions represented by programs? Although a high-level language such as Java is amenable to human programming, it's extremely sensitive to small changes in syntax and semantics. In genetic programming, solutions represented by programs are very sensitive to transformation operators. Thus, the appeal of specifying a solution as a program can be offset by the complexity of defining representation-sensitive transformation operators. In genetic programming, the difficulty in designing well-behaving operators is typically avoided by using random-transformation operators, carrying out a computationally complex search process, and tolerating overly complex solutions.

My PhD research addressed the need to understand the dynamics of genetic programming that encourage efficient, effective search. I focused on a critical property of genetic programming search: the population. This relates to many aspects of the genetic programming algorithm. Diversity was used to describe and analyze populations and their effect on search. The research led to several informative measures of diversity, useful for controlling and predicting the outcome of search, and algorithm enhancements that help explain the dynamics of genetic programming search on new problems. My future work addresses genetic programming's potential to leverage knowledge from other fields (such as programming languages and software engineering) to improve search (for example, to design better-behaving transformation operators and representations) and evolve better programs (that is, programs that are maintainable, extendable, and self-reconfigurable and that better use the vast array of existing software libraries).

My current work aims for innovation in science and algorithm research via real-world problem solving. At the GE Global Research Center, I research advanced learning and search algorithms for solving a wide range of data mining, optimization, and learning problems. Working in the scope of solving real-world problems requires balancing long-term research agendas with measurable short-term results. This creates the need to address often-overlooked research problems, such as the inefficiency of genetic programming transformation operators and solutions. Applying adventurous AI methods such as genetic programming to real-world problems has the potential to create new, innovative solutions as well as provide critical advancements to the method.