LAURENCE D. MERKLE

January 3, 2008

Work:
5500 Wabash Ave., CM-103
Rose-Hulman Institute of Technology
Terre Haute, IN 47803
812-877-8474 (voice)
812-872-6060 (FAX)
Laurence.D.Merkle@rose-hulman.edu
http://www.rose-hulman.edu/~merkle/

Home:
2122 N. 11th St.
Terre Haute, IN 47804
812-234-6536 (voice)
812-230-0316 (mobile)
l.merkle@ieee.org

ACADEMIC POSITIONS

Assistant Professor of Computer Science **2002-present**

Rose-Hulman Institute of Technology, Terre Haute, Indiana

Assistant/Associate Professor of Computer Science **1999-2002**

United States Air Force Academy, Colorado

Adjunct Faculty Member **1998-1999**

College of Santa Fe, Albuquerque, New Mexico

Adjunct Faculty Member **1997-1998**

Chapman University, Albuquerque, New Mexico

Adjunct Faculty Member **1997-present**

Department of Electrical and Computer Engineering
Air Force Institute of Technology, Wright Patterson AFB, Ohio

OTHER RELEVANT POSITIONS

Visiting Professor
Information Institute, Air Force Research Laboratory, Rome, New York **2004**

- Completed high-level design for an "Evolutionary Algorithm" core for use in Field Programmable Gate Arrays
- Mentored undergraduate summer hire in the implementation of farming model and island model parallel implementations of an evolutionary algorithm to solve the problem of parameter fitting for a set of nonlinear differential equations modeling an antigen-antibody binding process of interest in DARPA's bio-computation program
- Contributed to the "Polymorphous Computing Architectures" section of a joint DoD/NASA proposal for Congressional funding in advanced computing architectures research

Chief, Center for Plasma Theory and Computation (1998-1999) and
Leader, Computational Plasma Physics Group (1996-1998)
Air Force Research Laboratory, Kirtland AFB, New Mexico **1996-1999**

- Led 15 scientist team, with 10 Ph.D.'s and a budget of $1.1 million per year, developing and applying parallel computational plasma physics software for design of high-power microwave (HPM) devices

- Managed a $3 million per year HPCMO Computational Electromagnetics and Acoustics effort.
- Managed $2.7 million High Energy Theory and Experiment contract.
- Using an evolutionary algorithm identified a more effective, lighter, cheaper HPM source design.
- Designed and developed enhancements for parallel software tools for particle-in-cell (PIC), computational magnetohydrodynamics (MHD), and computational electromagnetics (CEM) simulations.
- Conducted computational PIC and MHD simulations of pulsed power devices and HPM sources using various scientific workstations and high performance scalable architectures.

Artificial Intelligence Project Officer
AI Program Management Office, Wright-Patterson AFB, Ohio **1988-1991**

- Promoted insertion of AI technologies into Air Force logistics processes.
- Taught 40-hour short courses on expert systems and M.1 programming to Air Force Logistics Command personnel.
- Designed and implemented an automated text-retrieval system prototype that has evolved into the Air Force Acquisition Management system.

Individual Mobilization Augmentee (“Category B” Air Force Reservist)
Air Force Office of Scientific Research, Arlington, Virginia **2002-2007**

- Allocate funds for the United States Air Force’s basic research program in High Performance Computing.
- Prioritize the United States Air Force’s requirements for Department of Defense High Performance Computing Program resources.

EDUCATION

Ph.D. in Computer Engineering 1996

Air Force Institute of Technology, Wright-Patterson AFB, Ohio
Dissertation: *Analysis of Linkage-Friendly Genetic Algorithms*
Minor: Biochemistry (Wright State University)
Chairperson: Gary B. Lamont, Ph.D.

M.S.C.E. (Master of Science in Computer Engineering) 1992

Air Force Institute of Technology, Wright-Patterson AFB, Ohio
Thesis: *Generalization and Parallelization of Messy Genetic Algorithms and Communication in Parallel Genetic Algorithms*
Advisor: Gary B. Lamont, Ph.D.

B.S. in Computer and Systems Engineering 1987

Rensselaer Polytechnic Institute, Troy, New York

ORGANIZATION OF CURRICULUM VITAE

TEACHING

Philosophy

The learning process always involves at least one person, the student, within whom some change of mental organization takes place. In some cases, the process also involves a second person, the teacher, whose role is to affect the student's environment in such a way as to allow the learning process to become more effective, more efficient, or both. This can be accomplished by fulfilling traditionally recognized "sage on the stage" teaching roles such as:

- presenting material from alternative viewpoints;
- clarifying the most difficult concepts;
- placing concepts within the context of the course, the curriculum, the profession, and society; and
- providing opportunities for students to demonstrate their mastery of the subject.

However, the importance of other roles should not be overlooked. For example, teachers can improve the learning process by designing and suggesting student activities that lead students toward independent knowledge discovery or the exchange of ideas between peers. Not only can these approaches result in more effective and efficient learning in a specific domain, but they leave the student better prepared to learn in the future.

One example of the teacher is the tutor, who has considerable time to devote to each student, and can therefore rely on intuitive and very effective techniques involving plenty of two-way communication. In contrast, most classroom instructors have many students, and thus do not have the luxury of abundant time. As a result, they have the responsibility of budgeting their time in such a way that they maximize their overall positive impact on the learning process for their many students (in some sense that cannot be defined precisely).

Historically, most university professors have responded by relying on lecture and the other traditional techniques listed above for their classroom instruction. These techniques have certain clear advantages, such as allowing professors to cover more material in each class meeting than less traditional techniques do. They also provide obvious avenues for professors to reinforce or supplement the textbook, as appropriate. Finally, generally speaking, professors are comfortable lecturing. However, it is now fairly well recognized that lecture alone also has limited impact on the learning process.

One reason for the limited effectiveness of lecture is that even good students have attention spans shorter than the typical class period. Another is that listeners retain little of what they hear (even within their attention span) unless they engage in other activities to reinforce what they have heard. Thus, professors who aspire to excellence in teaching must be willing to choose from a set of techniques that includes not only lecture, but also active and collaborative learning.

I have experimented with a number of innovative teaching techniques, some of which have been more successful than others, and some of which are better known than others. One well-known in-class exercise technique that I use regularly is to divide the class into small groups to work on a problem. At the end of such an exercise, I usually pick one group to present their solution, and then ask another group to comment on the solution. The ensuing discussion is almost always the most valuable part of the exercise. One of my favorite techniques involves multiple choice quizzes, and is less well-known. After first taking a quiz individually and turning it in, students take the same quiz working in small teams. Each student's overall grade on the quiz is a combination of their individual score and the team's score. The discussion during the team portion of the quiz naturally focuses on exactly those topics with which students are less confident, and is invariably energetic.

Teaching is like any other profession, in the sense that practitioners must be wiling to continuously improve their skills. For example, outstanding communication is prerequisite to effective lecturing, and to teaching in general. Professors must adapt their communication skills based on the learning styles and backgrounds of their audiences, among other factors. The only way to develop and maintain communication skills is through practice.

Teachers must also continuously improve their expertise in their subject areas, as well as their knowledge of innovative teaching techniques. This is especially true in the rapidly developing fields related to computing. Engaging in an active research and professional development program leads to greater depth and breadth of understanding, and has the added benefit that it provides real world examples for incorporation in classroom discussions.

Goals

When used wisely, computing technology has the potential to dramatically improve the quality of people's lives, but it also has equally dramatic potential for abusive or ill-considered application. As such, modern society is in desperate need of leaders who are prepared to make well-informed, ethical, and socially responsible decisions about the use of computing technologies. With this need in mind, my overarching goal as a teacher of undergraduate computer science and related topics (and my personal calling) is to do my part in preparing those leaders. In order to achieve this goal, I aim to prepare my students not only for their first jobs in the computing disciplines, but for successful professional careers and responsible participation in society. As such, I recognize the importance of the specific material in each course I teach, but I also believe that every college-level course should introduce students to new ways of thinking. I am also constantly looking for ways to encourage students to think about the social and ethical implications of the projects and technologies with which they work. Finally, I attempt to lead by example by studying broadly and leading a life of service.

Experience

My first academic appointment was a part time teaching opportunity in which I taught five courses in one year at the Albuquerque campus of Chapman University.[1] I was assigned to the Air Force Research Laboratory as a researcher and project manager at the time. When Chapman decided to restructure its system of satellite campuses, I helped arrange the details of a matriculation agreement with the College of Santa Fe. I taught three more courses for that institution before being reassigned to the United States Air Force Academy (USAFA).

At USAFA, I taught the standard load of three sections each semester for my first two years (with either two or three preparations each semester). My teaching experiences there culminated in my role as the Course Director (CD) for the *Introduction to Computing* course during my third year, which is the subject of a vignette several pages below.

In 2002, rather than accept an Air Force assignment that would have taken me out of both research and teaching, I left behind a successful 15-year military career in favor of my academic career. I chose a position at the Rose-Hulman Institute of Technology because of the outstanding students, the unambiguous focus on undergraduate education, the strong sense of community, and the proximity to my wife's family.

Over the time described above, I have taught the following courses:

- Artificial Intelligence
- Compiler Construction
- Computer Architecture I and II
- Computer Programming I and II
- Computer Systems Analysis and Design I and II
- Operating Systems
- Computer Security
- Data Structures
- Design and Analysis of Algorithms
- Fundamentals of Software Development I
- Great Principles in Computing
- Introduction to Computers and Data Processing
- Introduction to Computing
- Organizational Information Systems
- Theory of Computation

[1] My teaching experience began earlier. As a rising 8th grader, I gave a chemistry lecture and demonstration to several hundred high school students attending a summer program at the University of New Mexico. I tutored in mathematics and computer science during high school and as an undergraduate. My first Air Force assignment gave me the opportunity to teach short courses in the development of expert systems. In graduate school I gave guest lectures in algorithms and parallel computing.

I have also served as the advisor for four independent studies and eight senior theses, served as a committee member for 7 master thesis and 2 doctoral thesis committees, and taught portions of a Preparation for Fundamentals of Engineering Exam course.

Short Courses (Air Force Logistics Command)

"Introduction to Artificial Intelligence and Expert Systems" 1-week course for engineers and logisticians on practical applications of AI	1988-1991
"Introduction to M.1 Programming" 1-week course for the same audience in the use of a simple expert system shell	1988-1991

Guest Lectures

"Evolutionary Algorithms", Rose-Hulman Institute of Technology	2004, 2006, 2007
"Introduction to Messy Genetic Algorithms," University of New Mexico	1997
"Introduction to Genetic Algorithms," Air Force Institute of Technology	1993-1996
Minimum Spanning Tree Algorithms (2 lessons), Air Force Institute of Technology	1994
Chemistry "Magic Show" and lecture for New Mexico High School summer program at University of New Mexico	1979

Interests

I continue to be interested in teaching courses similar to those I have taught before, as well as some for which I have not yet had the opportunity. For example, I could teach the following with relatively little time to prepare:

- Calculus
- Differential Equations
- Digital Logic
- Discrete Mathematics
- Evolutionary Computation
- Linear Algebra
- Numerical Analysis
- Parallel Computation
- Probability and Statistics
- Programming Languages (survey of paradigms style)

Other courses of interest that would require more time for me to prepare include:

- Computational Science and Engineering
- Computer Networks
- Computing and Society
- Cryptography
- Current Research Topics in Computer Science
- Ethics in Computing
- Number Theory
- Problem Solving
- Programming Languages (design style)
- Quantum Computing

Rose-Hulman Institute of Technology

CSSE[2] 120 Fundamentals of Software Development I	Required for Computer Engineering, Computer Science, Electrical Engineering, Mathematics, and Software Engineering majors	Sp 2002-03: 2 sections Wi 2003-04: 2 sections Wi 2006-07: 1 section
CSSE 232 Computer Architecture I	Required for Computer Engineering, Computer Science, and Software Engineering majors	Fa 2002-03: 1 section Wi 2002-03: 2 sections Fa 2003-04: 2 sections Fa 2005-06: 1 section Wi 2005-06: 1 section Fa 2006-07: 1 section Wi 2006-07: 1 section Fa 2007-08: 2 sections
CSSE 332 Operating Systems	Required for Computer Engineering, Computer Science, and Software Engineering majors	Wi 2004-05: 2 sections Sp 2005-06: 1 section Sp 2006-07: 1 section Su 2006-07: 1 student
CSSE 442 Computer Security	Elective	Sp 2004-05: 2 sections Sp 2005-06: 1 section Sp 2006-07: 1 section Wi 2007-08: 1 section
CSSE 473 MA[3] 473 Design and Analysis of Algorithms	Fills theory elective for Computer Science majors and Computer Science elective for Mathematics majors	Fa 2006-07: 1 section Sp 2006-07: 2 students
CSSE 474 MA 474 Theory of Computation	Fills theory elective for Computer Science majors and Computer Science elective for Mathematics majors	Sp 2003-04: 2 sections Wi 2007-08: 1 section
CSSE 490 Great Principles in Computing	Optional for any major	Wi 2007-08: 1 section
CSSE 491 Directed Independent Studies	Optional for any major	Sp 2002-03: 1 student Sp 2004-05: 1 student Su 2004-05: 1 student Sp 2006-07: 1 student
CSSE 495/496/497 Senior Thesis I/II/III	Optional for Computer Science majors	Fa/Wi 2002-03: 1 student[4] Fa/Wi 2003-04: 1 student[5] 2004-05: 2 students[6] 2005-06: 2 students[7] 2006-07: 1 student 2007-08: 1 student
ECE[8] 332 Computer Architecture II	Required for Computer Science and Computer Engineering majors	Fa 2004-05: 1 section

---

[2] CSSE = Computer Science and Software Engineering
[3] MA = Mathematics
[4] The thesis student (Mike Simon) published a paper based on his thesis research in the Undergraduate Student Workshop of the 2003 Genetic and Evolutionary Computation Conference.
[5] The thesis student (Ryan Poplin) and I submitted a journal article based on his thesis research.
[6] One of the thesis students (Eric Borzello) and I coauthored a refereed conference paper based on his thesis research in the Proceedings of the 2005 Congress of Evolutionary Computation.
[7] One of the thesis students (Mike McClurg) published a paper based on his thesis research in the Undergraduate Student Workshop of the 2006 Genetic and Evolutionary Computation Conference.
[8] ECE = Electrical and Computer Engineering

United States Air Force Academy

Computer Science 471 Artificial Intelligence	Optional for Computer Science and Computer Engineering majors.	Fa 1999 and 2000: Instructor[9] and Course Director (CD)[10].
		Sp 2002: Curriculum Committee Representative (CCR)[11].
Computer Science 110 Introduction to Computer Science	Required for all cadets (approximately 30 sections per semester).	Fall 1999: Instructor of 1 section
		Sp 2000: Instructor of 2 sections
		Fall 2000 and Sp 2001: CCR, as well as instructor for 1 experimental section using Lego Mindstorms to teach programming concepts.
		Fall 2001: CD (17 instructors) and instructor of 1 section
Computer Science 110 Introduction to Computing		Sp 2002: CD (18 instructors) and instructor of 1 section
Computer Science 380 Algorithms and Data Structures	Required for Computer Science and Computer Engineering majors.	Sp 2000: CCR and 1 of 3 instructors.
		Sp 2001: CD (2 instructors) and instructor of 1 section.
Engineering 402 Preparation for Fundamentals of Engineering Exam	Encouraged for engineering majors.	Sp 2000, 2001, and 2002: Instructor for digital computing topics.
Computer Science 499 Independent Study	Optional for Computer Science majors.	Fa 2000: Mentor for 2 cadets investigating automatic domain decomposition using evolutionary algorithms.
Computer Engineering 465 Computer Systems Analysis and Design I	Required for Computer Engineering majors.	Fa 2000 (initial offering): CD and instructor.
		Fa 2001: CCR.
Computer Engineering 466 Computer Systems Analysis and Design II	Required for Computer Engineering majors.	Sp 2001(initial offering): CD and instructor.
		Sp 2002: CCR.

[9] USAFA instructor responsibilities include: preparing and delivering lectures; leading discussions; developing classroom and laboratory activities; assisting in development of handouts, homework assignments, programming exercises, exams, and other graded work; grade homework, programming exercises, exams, and other graded work.

[10] USAFA Course Directors (CDs) have overall responsibility for their courses, including design of syllabi; selection of textbooks; development of handouts, homework assignments, programming exercises, exams, and other graded work; development and maintenance of course website, and recommendation of course grades to the Dean.

[11] USAFA Curriculum Committee Representatives (CCR) within the Department of Computer Science provide final review and approval of all materials for their courses except textbooks, including syllabi, handouts, homework assignments, programming exercises, exams, and other graded work. Textbooks are reviewed and approved directly by the Curriculum Committee.

My teaching experiences at the United States Air Force Academy (USAFA) culminated in my role as the Course Director (CD) for the *Introduction to Computing* course (CS 110). CS 110 is a “core” course, meaning that it is a graduation requirement for every cadet at USAFA, regardless of their major. As such, over 1,000 cadets take it every year. The year that I directed the course, the enrollment exceeded 1,300 cadets in 61 sections. The course always comprises the majority of the department’s teaching workload.

As the CD, I had overall responsibility for the course, including design of the syllabus; selection of the textbook; development of handouts, homework assignments, programming exercises, exams, and other graded work; development and maintenance of the course website, and recommendation of course grades to the Dean. Course grades have special significance at USAFA, because GPA has a direct effect on the cadets’ first active duty assignments. Thus, as the CD, I was also responsible for ensuring consistency across instructors, including grading and disclosure of information about the graded work.

During the year prior to the Fall 2001 semester (my first as CD), I led the department through a ground-up redesign of the course. The remainder of this section describes the changes to the course and their results.

The primary purpose of the redesign was to make the course more relevant to cadets who do not declare computing-related majors. Previously, the course focused heavily on programming topics, and addressed other topics only at the knowledge and comprehension learning levels. This was appropriate for cadets who would eventually choose a computing-related major, but not for the vast majority of cadets enrolled in the course. After lengthy discussion, we decided to increase the emphasis on a number of non-programming topics, including algorithms, hardware, operating systems, networking, the World Wide Web, security, multimedia, databases, modeling, and simulation. We developed application-level learning objectives for each of these topics.

At the same time, we incorporated a number of other changes to improve the effectiveness of the course for all cadets:

- We organized the course into “blocks” of lessons. We had previously had this structure only for programming topics.
- We added web-based pre-assessment quizzes covering the reading for each block at the knowledge and comprehension learning levels.
- In the process of redeveloping the lesson plans, I replaced lectures with active and collaborative teaching techniques wherever I could. This was particularly effective in combination with the pre-assessment quizzes.
- Each incoming USAFA class buys standard issue computers. The Class of 2005 was the first class to receive notebook computers. We required them to bring their computers to class, and I incorporated their use in a majority of the lesson plans.
- At the time, USAFA used Ada 95 as the high-level programming language for the course because it is very easy to learn. Nonetheless, it has a few syntactic structures that are unnecessarily complicated for an introductory course (e.g. declaration of an array variable requires a type definition). We developed a “CS110” package that simplifies the syntax (e.g. by providing standard array types) to allow us to focus on the principles behind the programming constructs.

- There is great variety in the backgrounds of incoming cadets. As such, in the past, a few cadets in each section were not challenged by the course. We instituted an Honors version to challenge these more advanced cadets. The Honors version covered the same topics as the regular version in greater depth, as well as additional programming topics.

These changes produced outstanding results. Most importantly, instructors felt that their cadets learned more than in previous semesters. Objectively, the cadets scored better on exams, supporting this belief.

The pre-assessment quizzes had the intended effect: the cadets were better prepared for class, so we could use class time to focus on the more difficult application level learning objectives. The drawback was that our first implementation of the quizzes was based on a free plug-in for Microsoft FrontPage, which turned out to somewhat inflexible and unreliable. We later developed our own implementation that worked better for us.

Finally, despite the course's bad reputation, it received outstanding student critiques. Specifically, cadets gave it one of the lowest ratings at USAFA for the statement "Prior to taking this class, I was interested in the content of this course," but at the end of the semester, they rated it the highest out of the 11 core courses in the Basic Sciences Division and the Engineering Division in 9 of 36 categories:

- Intellectual challenge and encouragement of independent thought
- Evaluative and grading techniques (tests, papers, projects, etc.)
- The course as a whole
- Amount I learned in the course
- This course improved my ability to deal with problems that don't have an approved solution
- My motivation to learn has increased because of taking this course
- There are a number of things in this subject I'd like to learn more about
- My instructor designed activities that made me think
- I believe all the information contained in this critique is anonymous

College of Santa Fe[12]

Computer Science 230 Computer Programming I	Required for Computer Science majors.	Term 3 1998: Full responsibility for all aspects of course.
Computer Science 231 Computer Programming II	Required for Computer Science majors.	Term 4 1998: Full responsibility for all aspects of course.
Computer Science 350 Data Structures	Required for Computer Science majors.	Term 4 1998: Full responsibility for all aspects of course.

Chapman University

Computer Science 200 Intro to Computers and Data Processing	Required for Computer Science majors.	Term 3 1997[13]: Full responsibility for all aspects of course.
Computer Science 402 Compiler Construction	Required for Computer Science majors.	Term 3 1997: Full responsibility for all aspects of course.
Computer Science 350 Data Structures	Required for Computer Science majors.	Term 4 1997, Term 3 1998: Full responsibility for all aspects of course.
Computer Science 390 Artificial Intelligence	Optional for Computer Science majors.	Term 5 1997: Mentor for one student.
Computer Science 315 Organizational Information Systems	Required for Computer Information Systems majors.	Term 1 1998: Full responsibility for all aspects of course.

[12] Chapman University closed its Albuquerque Academic Center in 1998. Under an articulation agreement, the Albuquerque campus of the College of Santa Fe offered Chapman University courses to allow students to complete their degrees. Thus, the Term 3 1998 and Term 4 1998 were Chapman University courses, although they were taught at the College of Santa Fe, and I was officially an adjunct faculty member of the College of Santa Fe.

[13] The academic year for the Albuquerque Academic Center of Chapman University consisted of five terms of nine weeks. All of the courses that I taught had five contact hours per week.

PROFESSIONAL DEVELOPMENT

Research

Within the context of the teaching loads at the United States Air Force Academy (USAFA) and the Rose-Hulman Institute of Technology, I am a relatively active researcher. In my last year at the USAFA, I received the Outstanding Researcher award for the Department of Computer Science. Also, until recently, I was the only member of the Rose-Hulman Department of Computer Science with an externally funded research program (excluding grants for instructional technology).

Many of my research contributions over the past 15 years have involved the effective and efficient application of evolutionary algorithms as optimum seeking techniques. This work has included theoretical analysis, often supported by computational experiments, as well as practical applications in the solution of important scientific, engineering, and mathematical problems. As such, it has been highly interdisciplinary and collaborative.

My position at Rose-Hulman in particular has given me a number of opportunities to establish mentoring relationships with undergraduates involved in various research activities. In particular, in my sixth year at Rose-Hulman, I am advising my eighth senior thesis student. All of my thesis students have chosen their own research topics. Four of them have involved various aspects of evolutionary computation (one application in machine learning, another in multi-agent cooperation, a third in searching for metabolic pathways, and a fourth in traffic engineering). Two others have been at the intersection of computer science and discrete mathematics (one involving algorithms for visualizing and analyzing knots, another in the complexity analysis of an abstract game). My previous thesis student was interested in user interface design, and chose to systematically assess the interfaces of several statistical analysis tools commonly used in science and engineering. My current thesis student is investigating computer security with a focus on critical infrastructure systems. Three of my thesis students have presented their work at premier evolutionary computation conferences.

My engagement of undergraduates in research has also included advising three independent study students. Most notably, though, I actively involved three undergraduates as research assistants in my externally funded research on Evolutionary Computation in Polymorphic Computing Architectures. All of them have made important intellectual contributions to the work and have been included as coauthors on workshop presentations.

I have also been active in graduate education through my service on thesis committees. Between 1999 and 2003, I served on the committees of five Air Force Institute of Technology graduate students. Three were at the master's level and two at the doctoral level, all in the area of evolutionary algorithms. Both of the dissertations were in some sense descended from my own, and as such I played very active roles. Although my department at Rose-Hulman has no graduate students and no graduate courses, faculty in other departments have sought me out to serve on the committees of their masters students, and I have served on four such committees.

Theory of Evolutionary Algorithms

While other commonly used optimum seeking techniques require continuity, differentiability, convexity, monotonicity, or other mathematical properties of the objective function(s), evolutionary algorithms require none of these. The obvious advantage is that they can be applied effectively to a very broad set of problems. The disadvantage is that because they ignore these mathematical properties even when they are present, evolutionary algorithms can be very inefficient. Fortunately, experience with a large number of real-world applications has shown that the efficacy of evolution-based search can be improved greatly by combining the evolutionary algorithm with efficient local optimization techniques. The resulting "hybrid" or "memetic" algorithms are more efficient than the evolutionary algorithm alone and more effective than the local technique alone. Another empirical observation regarding evolutionary algorithms is that they tend to scale better with respect to dimensionality than other commonly used optimum seeking techniques.

Analysis of parallel tournament selection

The messy genetic algorithm was proposed in order to address perceived limitations of the simple genetic algorithm associated with the "static building block hypothesis." The algorithm's run time is dominated by the "primordial" phase, in which tournament selection is used to reduce the population size from a combinatoric function of the problem size to a polynomial function. As part of my masters thesis research, I proved that the underlying algorithm is "inherently sequential," and then identified and analyzed four variations in terms of their speedup in multiprocessor environments (Merkle and Lamont, 1993). My computational experiments were consistent with my theoretical results and demonstrated excellent parallel efficiency with no statistically significant difference in algorithmic effectiveness in most cases.

Future directions

Although the messy genetic algorithm is no longer of practical interest, tournament selection is used in many current evolutionary algorithms because of its translation and scale invariance. As such, I plan to adapt my analysis of parallel tournament selection to apply to broader classes of algorithms.

"Scheduling" for the fast messy genetic algorithm

After successfully analyzing the parallelism present in the messy genetic algorithm, I turned my attention to developing a probabilistic model of the effectiveness of various parallel mappings of its successor – the "fast messy genetic algorithm" (fmGA). The problem turned out to be much more difficult than the previous one because of the importance of second order statistical effects in determining the dynamic behavior of the algorithm. At the time I lacked the mathematical maturity for a complete analysis, and consequently the published results were incomplete. However, the experience sowed the initial seeds of my intellectual independence from my advisor, and in hindsight was a major step towards the identification of my dissertation topic.

That topic was the analysis of the class of linkage-friendly genetic algorithms (lfGAs), which consists of EAs that use strictly invariant selection operators and order invariant representation schemes. Sensitivity of lfGA effectiveness to exogenous parameters limits their practical application. As such, I chose the topic with an eye towards exogenous parameter selection ("scheduling") for fmGAs, which are lfGAs that use binary tournament selection (BTS) with thresholding, periodic filtering of a fixed number of randomly selected genes from each individual, and generalized single-point crossover. I proposed probabilistic variants of thresholding and filtering, and defined EAs using the probabilistic operators to be generalized fmGAs (gfmGAs).

I also developed a dynamical systems model of lfGAs that permits prediction of expected effectiveness. BTS with probabilistic thresholding is modeled at various levels of abstraction as a Markov chain. Transitions at the most detailed level involve decisions between classes of individuals. The probability of correct decision making is related to appropriate maximal order statistics, the distributions of which I obtained. I extended existing filtering models to include probabilistic individual lengths.

With the groundwork laid, I posed the lfGA parameter selection problem formally as a constrained optimization problem in which the cost functional is related to expected effectiveness and derived Kuhn-Tucker conditions for the optimality of gfmGA parameters. Finally, I proposed parameter selection techniques for both fmGAs and gfmGAs.

Future directions

Exogenous parameter selection is arguably the most significant obstacle to the effective use of evolutionary algorithms. Fortunately, techniques similar to those applied in this research can be applied to a broad class of those algorithms to address that problem.

Formalization of the class of evolutionary algorithms

There is not a universally accepted definition of the term "evolutionary algorithm." Loosely speaking, it refers to a member of the class of population-based stochastic algorithms involving the iterative application of selection, recombination, and mutation operators inspired by the processes of biological evolution. In order to convey the idea more precisely to the mathematician on my dissertation committee, I proposed a formal framework for EAs in which evolutionary operators are viewed as mappings from parameter spaces to spaces of random functions (Merkle and Lamont, 1997). Formal definitions within this framework capture the distinguishing characteristics of the classes of recombination, mutation, and selection operators. This is the most precise yet inclusive definition of the term "evolutionary algorithm" of which I am aware.

Future directions

First, an obvious but necessary generalization of the framework is in order to allow for the possibility of multiple objective functions. More importantly, the existence of a commonly accepted rigorous definition would facilitate the maturation of evolutionary computation theory. Currently, results are proved and stated as if they apply to all evolutionary algorithms, whereas the authors' unexpressed notions of that class are not necessarily the most general. Conversely, results that have been proved for one class of evolutionary algorithms are proved again for other classes that turn out to be special cases of the first class. I envision demonstrating the generality of the random function framework by producing a collection of both new and existing theoretical results using it as the common starting point.

Finally, the proposed framework could form the basis of a Computer Aided Software Engineering (CASE) tool to assist in the development of application-specific evolutionary algorithms. Initially, such a tool would be useful primarily in the sense of reducing software development time for evolutionary computation practitioners who have the "expert knowledge" necessary to develop the systems. However, in the long term, it could include heuristic rules to assist experts in other fields to make effective use of state-of-the-art evolutionary computation techniques.

Applications of Evolutionary Algorithms in Science, Engineering, and Mathematics

Many important problems in science, engineering, and mathematics can be cast as optimization problems. While some are amenable to calculus-based optimization or other standard techniques, many are not because the associated mathematical models lack the necessary properties mentioned above. Accordingly, many of these problems are appropriate applications for evolutionary algorithms as optimum seekers. Interestingly, while these applications often result in the desired solutions to the original problems, they also tend to explore areas of the search space that are not modeled accurately, thereby identifying fruitful refinements of the mathematical models.

This line of research has afforded me valuable opportunities for collaboration with experts in a number of areas outside of computer science. In particular, my sustained work on polypeptide structure prediction has given me the opportunity to learn considerable polymer chemistry. Similarly, my involvement in microwave source design has led me to learn about electromagnetics and plasma physics. These two problems are discussed in some detail below, along with my more recent work at the intersection of evolutionary computation and polymorphic computing architectures. My shorter term projects, many of which have also involved fruitful collaborations, have included a proposal for an EA-based network intrusion detection system (Merkle, Carlisle, Humphries, and Lopez, 2002), an EA-based automated load-balancing algorithm for a distributed discrete-event simulation (Carlisle and Merkle, 2004), and a swarm-based algorithm for multi-agent cooperation (Borzello and Merkle, 2005)

Polypeptide Structure Prediction

The "protein folding problem" is well known. A general "solution" to this problem would provide a general and efficient capability to predict accurately a polypeptide's tertiary structure given (only) its amino acid sequence. Such a capability would find immediate application in numerous important scientific, medical, and engineering applications. For example, knowledge of the tertiary structure of an enzyme is fundamental to understanding the mechanism by which it functions. Also, without this capability, *ab initio* design of new polypeptides with specific biochemical, mechanical, or optical properties (the "inverse protein folding problem") is infeasible.

Hybrid evolutionary algorithm

One common approach to the problem involves the minimization of the free energy of the polypeptide as a function of its conformation. Of course, this is a challenging optimization problem because the conformation space has high dimensionality. Furthermore, the energy landscape has many local minima and many singularities. During my PhD program, though separate from my dissertation research, I made major contributions to research involving the application of various evolutionary algorithms in such an energy minimization approach. Most notably, I proposed a hybrid genetic algorithm that incorporates efficient gradient-based minimization directly in the fitness evaluation (Merkle, Gates, Lamont, and Pachter, 1996). The algorithm includes a replacement frequency parameter which specifies the probability with which an individual is replaced by its minimized counterpart. Thus, the algorithm can implement either Baldwinian, Lamarckian, or probabilistically Lamarckian evolution.

My coauthors and I empirically evaluated the effectiveness of several variations of the algorithm by applying the techniques to the minimization of the polypeptide-specific ECEPP/2 energy model. The target molecule was the pentapeptide [Met]-Enkephalin, which was the *de facto* standard test for such algorithms at the time. The probabilistically Lamarckian variations obtained significantly better energies than did the standard algorithm followed by local minimization, supporting earlier results suggesting that the local minima in the energy landscape of [Met]-Enkephalin occur somewhat regularly. Also, the final energies obtained by the probabilistically Lamarckian algorithms using fitness proportionate selection were significantly lower than those obtained using tournament selection. The latter converged prematurely, suggesting that the selective pressure of standard binary tournament selection is too high for this application. Finally, the fitness proportionate selection runs obtained the global minimum significantly more frequently than the tournament selection runs. One of the other runs identified a unique conformation with an energy within 0.001 kcal/mol of the global minimum and having an rms deviation of 2.589 Å relative to the accepted global minimum. This last result was a success in the sense that the algorithm was very effective in optimizing the objective function to which it was applied. However, it also highlighted questions regarding the accuracy of the energy model and the viability of the energy minimization approach, since the identified conformation had not been observed experimentally.

Incorporating domain-specific knowledge

When I graduated from my PhD program, my advisor and I agreed that the greatest potential for improved effectiveness of evolutionary algorithms for energy minimization was through the incorporation of domain-specific knowledge in the search process. As such, one of his masters students proposed constraining the independently variable dihedral angles to values “allowed” by a linear approximation to the standard Ramachandran plot. Although the effectiveness of the proposed algorithm was worse in a statistical sense than that of the algorithm described above, it did identify conformations for which the CHARMM energy model predicted lower energies than that of the commonly accepted minimum energy conformation.

Future directions

Considerable progress has been made in this field in the time since I was last actively involved, but many challenges remain. In particular, alignment to known template structures was a relatively new idea at the time, and has now become a standard part of the approach to the problem. At the "easy" end of the difficulty spectrum, corresponding roughly to "comparative modeling" problems, existing alignment algorithms tend to find alignments very near the best possible. However, their accuracy "falls steadily and approximately linearly with increasing target difficulty, but with a smaller slope than that of the fall-off in alignment accuracy." (A. Kryshtafovych, C. Venclovas, K. Fidelis, and J. Moult, Progress Over the First Decade of CASP Experiments, PROTEINS: Structure, Function, and Bioinformatics Suppl, 7:225-236, 2005). Thus, at the "hard" end of the difficulty spectrum, "new folds" remain a particular challenge.

Fortuitously, because the sponsor of my graduate research was interested in the design of macromolecules with specified nonlinear optical properties rather than the analysis of naturally occurring proteins, my research was aimed directly at the "new folds" aspect of the problem. Furthermore, template alignment is fundamentally an optimization problem, and ripe for the application of hybrid evolutionary algorithms. Finally, I remain convinced that the incorporation of domain-specific knowledge in the search process is essential to achieving greater effectiveness, but I believe that biased representation schemes and operators have greater potential for success in this domain than hard constraints.

Microwave Source Design

The Relativistic Klystron Oscillator (RKO) is a high-power microwave (HPM) source in which a high-energy annular electron beam passes through a cylindrical waveguide having two longitudinally separated annular cavities (Hendricks, Coleman, Lemke, Arman, and Bowers, Physical Review Letters, vol. 76, no 154, 1996). The transverse motion of the beam is restricted by a static magnetic field. An electron density oscillation is induced on the beam by an externally-driven low-power electromagnetic oscillation in the upstream ("driver") cavity. The position of the downstream ("booster") cavity is chosen so that the system has a resonant electromagnetic frequency that is excited by the oscillations in the electron beam, and thus the kinetic energy of the beam is converted into electromagnetic energy.

Because the electron beam is generated by the sudden release of a large but finite amount of energy, it is important to maximize the efficiency of energy conversion in the RKO. Existing devices have good steady-state efficiency, but take too long to reach steady-state. Thus, there is considerable interest in reducing the start-up time by improving the signal growth rate. Together with my collaborator, I have addressed this problem using a real-valued constraint-handling evolutionary algorithm based on the publicly available GENOCOP software (Merkle and Luginsland, 2003).

Our initial work relied on a circuit model that related the growth rate to the beam voltage, beam current, and gap separation. Optimization of the resulting function via analytical and standard numerical techniques is intractable because of the existence of many local optima, hence the decision to apply an evolutionary algorithm. The algorithm identified designs with growth rates that are significantly higher than previously investigated designs.

We evaluated the best of those designs using particle-in-cell (PIC) simulations, and determined that they are adversely affected by non-linear phenomena, such as virtual cathode formation. This led us to refine our mathematical model by incorporating non-linear constraints corresponding to the limiting current and the cutoff frequency. At the same time, we replaced the circuit model by a dispersion relation model, which accurately models both resonant frequencies of the system. Furthermore, it generalizes readily to a multi-cavity version of the RKO, suggesting a novel class of devices. However, evaluation of each candidate design involves calculating the characteristic polynomial of the system and then finding the roots of that polynomial. Fortunately, because of the special form of the matrix describing the system, there is an O(n) algorithm for calculating the characteristic polynomial.

Computational experiments identified multi-cavity versions of the device that offer signal growth rates an order of magnitude better than those of the best known 2-cavity versions. However, there is still room for improvement. The best multi-cavity designs identified by independent computational experiments tended to be dissimilar, suggesting that the identified designs were still far from the global optimum. Thus, we sought to improve the effectiveness of the EA, which we did by incorporating more sophisticated constraint-handling techniques. Rather than simply assigning zero fitness to infeasible individuals, we implemented versions of the algorithm that included a repair mechanism in the evaluation function as well as a feasibility-preserving recombination operator. Various combinations of these techniques have resulted in designs with even better signal growth rates, although the interactions of the techniques appear to be complex and somewhat non-intuitive.

Future Directions

Independent computational experiments still tend to identify dissimilar designs, suggesting that improving the effectiveness of the algorithm might lead to the identification of higher growth rate designs. However, as mentioned above, the results of the computational experiments involving constraint handling techniques are difficult to interpret. This is in part because the EA software used provides little diagnostic output. Future research will involve the modification of the software to provide fitness statistics and diversity measures. Aside from refining the constraint handling techniques, it is likely that effectiveness can be improved through the use of other memetic techniques (Lamarkian, etc.) and hybridization with local search (e.g. conjugate gradient).

Improved efficiency of the algorithm is another potential research topic. A farming model parallel version of GENOCOP is under development. Also, the efficiency of the fitness evaluation itself would be improved if a technique could be developed to reduce the number of roots found. The Lehmer-Schur algorithm is a promising candidate.

Once we are satisfied with the designs identified by the algorithm, those designs will be evaluated using a high fidelity PIC simulation. Those simulations might suggest a physical experiment, or they might suggest refining the mathematical model. Several avenues for improvement of the latter are available. One is to consider the limiting currents at cavity gaps. Another is to consider mode competition and sensitivity to design parameters.

Polymorphic Computing Architectures

As a visiting professor with the Advanced Computing Architectures Branch of the Air Force Research Laboratory, I focused on the development of a VHDL description of an Evolutionary Algorithm "core" for use in FPGA-based systems. I also laid the groundwork for a recently completed externally funded research project on Evolutionary Computation in Polymorphic Computing Architectures. This section describes that project.

Current computing systems are designed to support fixed, idealized application loads, and their performance inevitably suffers when the actual load doesn't match the idealized load for which they were designed. Also, as manufacturing processes for integrated circuits advance and we approach the fundamental limitations of silicon technology, wire delays are becoming more significant relative to gate delays. Polymorphic computing architectures (PCAs) represent a revolutionary approach to computing systems that seeks to provide processing capabilities that are both amenable to dynamic optimization as the application load changes and scalable with technology advances. Leading efforts in PCA research include the Raw microprocessor under development at the Massachusetts Institute of Technology and the TRIPS architecture project at the University of Texas at Austin. Both of these efforts achieve dynamic responsiveness and scalability through the use of tile-based architectures of some variety.

MIT researchers argue that we must reconsider our idea of machine instructions to include signal routing information along with the usual functional unit control information. The Raw microprocessor makes this possible, and has been demonstrated to provide two orders of magnitude better performance than traditional processors on certain applications. However, optimization of the routing information places an additional burden on the compiler. Compiler enhancements implemented shortly before the beginning of this project resulted in code with speed and tile usage that came close to hand-customized code, but independent evaluations resulted in only two-thirds of the theoretically possible efficiency, suggesting that further optimization was possible.

Researchers at the University of Texas at Austin also suggest a new paradigm for machine instructions, illustrated by their Tera-op Reliable and Intelligently Adaptive Processing System (TRIPS) architecture. They advocate the adoption of Explicit Data Graph Execution (EDGE) architectures, in which "the hardware delivers a producer instruction's output directly as an input to a consumer instruction," thereby eliminating most of the expensive logic that has found its way into architecture design over the past two decades. In addition to the usual requirements, a compiler targeting such an architecture must be able to identify blocks of instructions containing no branches, map each block to a tile for execution, and then map each operation in the block to a processing element. These spatial scheduling mappings affect both concurrency and communications delays, and thus result in a difficult multicriteria optimization problem. When this project began, the TRIPS compiler employed a greedy approximation algorithm. The resulting code was highly un-optimized and bloated, which suggested an opportunity to improve performance via a variety of optimization techniques.

The specific goals stated in the proposal for this effort were to explore the use of EC techniques that enable and are enabled by PCA technology:

- We will develop versions of both the Raw and TRIPS compilers that combine EC techniques with the compilers' existing algorithms. EC techniques provide robust global search, so they will be used to explore the space of schedules and choose areas in which to perform local search. The existing algorithms will provide the required efficient local search.
- Because of their population-based nature, EC techniques are amenable to a rich variety of implementations on parallel and distributed architectures and scale very well with processor count. Much of the research in this area will carry over directly to their implementation on tile-based polymorphic computing architectures. As such, EC techniques could be distributed spatially across a tile-based architecture to provide dynamic performance optimization. We will design, model, and evaluate the performance of both island model and farming model parallel EC implementations for both the Raw and TRIPS architectures. We will also evaluate the performance of those implementations empirically using a prototype board available to us from the MIT researchers and simulators for the other two architectures.

Early in the project, it became apparent that it was not feasible to enhance the scheduling by the Raw compiler because the burden of program decomposition and mapping currently rests on the programmer. Thus, the remainder of the effort focused on the TRIPS architecture.

Four general techniques were identified for the application of EC in enhancing compiler effectiveness. The Finch Meta-optimization Framework implementation of the compiler-algorithm-time technique was adopted for this effort. Other techniques include compiler-parameterization-time, compile-time, and schedule-time. Tradeoff considerations among these techniques include their impact on execution time, compilation time, compiler construction time, reproducibility of execution, reproducibility of compilation, and breadth of application space targeted.

Given the advantages and disadvantages of each of the techniques, various combinations are appropriate in specific situations. One of the motivations for PCAs is to achieve near-optimal performance on each mission-critical application in a dynamic workload. For such an application, it is reasonable to assume that it is worthwhile to invest considerable offline computational effort in order to obtain improvements in online performance. As such, the compiler-algorithm time, compiler-parameterization time, and compile-time techniques should all be considered. Furthermore, assuming that the set of critical applications in the workload is small, the compiler-algorithm time and compiler-parameterization time techniques are especially applicable, since their application tailors the compiler to the applications in the training set, which can be chosen to consist of exactly the applications of interest. The schedule-time technique also has potential applicability in the context of PCAs, but execution time predictability must be addressed before it is practical for use in operational environments.

In order to integrate the Finch Metaoptimization Framework with the Scale compiler used in the TRIPS toolchain, several modifications to the compiler were implemented. Immediately after parsing its command line arguments, the modified version of Scale invokes a Finch method that prepares for the evaluation of a candidate priority function generated by the evolutionary algorithm. Also, the method that is normally used to compute the priority function that is built into Scale was modified to instead invoke a second Finch method that evaluates the candidate priority function. Finally, immediately before terminating, the modified version of Scale invokes a third Finch method to finalize the evaluation of the candidate priority function.

Computational experiments were performed to evaluate the effectiveness of Finch in evolving in-lining priority functions for Scale. The experiments were executed on the Rose-Hulman Institute of Technology Beowulf cluster. This required porting both the TRIPS toolchain and Finch to the cluster, as well as modifying Finch’s interprocessor communication to make use of the Message Passing Interface (MPI) standard. The average number of instructions per generated hyperblock was used as the primary metric for these experiments, based in part on observations by the TRIPS developers that maximizing this metric is essential to achieving good performance. A number of applications were considered as possible inputs to Scale for the experiments. The “Ground Moving Target Indicator (GMTI),” was chosen because for the unmodified version of Scale the chosen metric varies gradually with the allowable code bloat parameter. Using small population sizes and small generation counts, the software occasionally obtains values of the metric equal to that produced by the unmodified compiler, but not reliably. Each experiment requires between three and four hours of wall clock time using 17 processors.

Future Directions

The limited success of the computational experiments described above should be interpreted in light of the fact that by genetic programming standards, the population size and generation count for these experiments are both extremely small. It is likely that larger values of either parameter would result in the identification of more effective in-lining priority functions. Furthermore, each experiment required less than four hours to execute, so using larger population sizes and generation counts would not result in prohibitive execution times.

This effort has laid the groundwork for the development of hybrid evolutionary algorithms that exploit both the global search properties of evolutionary computation and the effectiveness of the existing compiler optimization algorithms. Future research is needed in a number of areas:

- Perform additional computational experiments related to TRIPS in-lining, as well as similar experiments for other compiler optimizations involving priority functions (e.g. loop unrolling). These experiments can be completed without further modification of Finch. The advantage of those kinds of optimizations is that they have relatively direct impact on the formation of hyperblocks (which is where the greatest impact on performance can be made). The limitation is that they explore relatively small parts of the space of assembly language programs.

- Explore larger areas of the space of TRIPS assembly language programs by modifying Scale so that a Finch-optimized priority function controls the building of hyperblocks. This could be done in a few different ways. The most promising of these is modifying the control flow graph (CFG) creation function so that it consults the finch-optimized priority function. This would allow Finch to change the CFG so that it will make "better" hyperblocks, since the fitness function uses a heuristic that only takes into account the average number of instructions per hyperblock.
- The spatial distribution of an EA across a tile-based architecture to provide dynamic performance optimization still merits investigation.

Computer Science Education

While teaching at the United States Air Force Academy, I was involved in the assessment of the use of robots in an Introduction to Computing course. One of my colleagues developed a novice-friendly interface to the Lego Mindstorms RCX based on a subset of Ada that he calls Ada/Mindstorms 2.0. We developed laboratory exercises based on this interface that gave the students experience with sequential control, variables, constants, procedural abstraction, selection, iteration, and arrays. The excitement in the classrooms during these laboratories was very gratifying. Unfortunately, a careful statistical analysis revealed that the use of the robots did not improve the students' performance on exams, nor their likelihood to choose a computing-related major (Fagin and Merkle, 2002). Feedback from focus groups suggests that the unavailability of the robots outside of laboratory periods limited their effect on student learning.

Computers in Education

Since 1995, students at the Rose-Hulman Institute of Technology have been required to purchase an institute-specified laptop computer. Thus we understand and use the power of laptops in education. However, we also recognize that emerging technologies can still contribute to student learning. In particular, pen-based computing technologies such as tablet PCs and slate devices have the potential to improve visual communication in the classroom, as well as to facilitate collaboration when used in connection with appropriate software such as DyKnow Vision. As such, a number of my colleagues across the Institute have been experimenting with these technologies in their classes since 2004, and we now have two classrooms that are correspondingly equipped for both instructors and students. Some faculty use the pen-based technologies in other classrooms as well.

I have been using the technologies in my classes since 2005. My experience using it in Computer Architecture I is somewhat representative. The course is required for majors in computer science, software engineering, and computer engineering. I decided to implement DyKnow for several reasons. First, I believed DyKnow would help improve the student learning experience by facilitating more effective note-taking. Second, I thought it would help me obtain and provide feedback on student learning. Finally I believed it would help increase classroom interaction.

At this point in my experience with DyKnow, I know that at least some students take advantage of the ability to take notes on my DyKnow panels. My best evidence is the groaning I hear when I move things around on my panels (and I'm learning to avoid doing that). I've seen more note-taking in recent terms with the slate devices than in previous terms with either tablet PCs or no pen-based computing. DyKnow use has also helped me measure student learning more immediately in the classroom. Just asking the students "Do you understand" doesn't result in quality feedback. Usually, neither does inventing a simple problem on the spot. However, when I integrate feedback opportunities into my lesson plans, I get good feedback. For example, using DyKnow's survey feature, I ask multiple choice questions about the effects of code snippets to determine the degree to which students have learned the semantics of assembly languages. Also, using the panel submission feature I have the students anonymously submit short assembly language programs and then critique the submissions (with student help). Finally, I have recently started using the feature that allows me to provide direct feedback to students by collecting panels they submit during class and then annotating and returning them outside of class. The initial feedback from students is that this is very helpful.

It is clear to me that DyKnow changes classroom interaction, but it is also clear to me that care and practice are critical to ensuring that change is a positive one, since initially I spent more time looking at survey results and less time looking at the students. However, the educational benefits of the technology were sufficient to motivate me to continue experimenting with its use, and I am now seeing tangible benefits.

Publications and Presentations

Pending and in Review

L. D. Merkle. VHDL Implementation of a General Purpose Evolutionary Computation Core. In preparation for submission to IEEE Transactions on Evolutionary Computation.

L. D. Merkle, M. G. Ellis, and M. C. McClurg. Evolution of TRIPS Compiler Optimization Priority Functions. In preparation for submission to IEEE Transactions on Evolutionary Computation.

A. Chidanandan and L. D. Merkle. Use of Version Control Software in a Project-Based Introductory Computer Architecture Course. In preparation for submission to Computers in Education Journal.

Invited

L. D. Merkle. Selected Applications of Evolutionary Computation in Computational Science and Engineering. Indiana State University, Department of Life Sciences, 2006.

L. D. Merkle. Electronic Voting. Invited panel member, Illiana Information Technology Association October Meeting, 2004.

L. D. Merkle and J. W. Luginsland. Design Optimization for a Novel Class of High Power Microwave Sources. Proceedings of the 2003 IEEE Congress on Evolutionary Computation, presented in the special session on Evolutionary Design Optimization.

L. D. Merkle. Design Optimization for a Novel Class of High Power Microwave Sources: Incorporating Constraints in a Real-Valued Evolutionary Algorithm. Colloquium presented at various academic institutions, 2001-2002.

G. B. Lamont and L. D. Merkle. Towards Effective Evolutionary Algorithms for Polypeptide Structure Prediction. In G. Fogel and D. W. Corne, editors, Evolutionary Computation in Bioinformatics, 2003.

L. D. Merkle. Fielding expert systems: Really getting them used. Invited panel member, First World Congress on Expert Systems, 1991.

Journal Articles

J. Holden, R. Layton, L. Merkle, and T. Hudson. Underwater Hacker Missile Wars: A Cryptography and Engineering Contest. Cryptologia, Vol. 30, 2006, pp 69-77.

M. C. Carlisle and L. D. Merkle. Automated Load Balancing of a Missile Defense Simulation Using Domain Knowledge. Journal of Defense Modeling and Simulation, Vol. 1, Issue 1, April 2004, pp. 59-68.

B. S. Fagin and L. D. Merkle. Quantitative Analysis of the Effects of Robots on Computer Science Education. ACM Journal of Educational Resources in Computing, Vol. 2, No. 4, December 2002, pp. 1-18.

Conferences (Refereed)

A. Chidanandan, R. DeVasher, P. Ferro, D. Fisher, S. Mitra-Kirtley, L. Merkle, D. Mutchler, S. Sexton, M. Simoni, and J. Williams. Work in Progress – Assessing the Impact of Pen-Based Computing and Collaboration-Facilitating Software in the Classroom. Proceedings of Frontiers in Education 2007. To appear.

A. Chidanandan, J.P. Mellor, and L. D. Merkle. Design and Implementation of a Minuscule General Purpose Processor in an Undergraduate Computer Architecture Course. 2007 International Conference on Microelectronic Systems Education, 2007.

A. Chidanandan, R. DeVasher, P. Ferro, D. Fisher, S. Kirtley, L. Merkle, D. Mutchler, M. Simoni, S. Sexton, A. Watt, and J. Williams. Evaluating the Symbiosis of DyKnow Software and Pen-Based Computing in the Rose-Hulman Classroom. The Second Workshop on the Impact of Pen-based Technology on Education, 2007.

Z. Chambers, A. Chidanandan, R. DeVasher, L. Merkle, M. Minster, S. Mitra-Kirtley, D. Mutchler, S. Sexton, A. Watt, J. Williams, and M. Zoetewey. What is Beyond the Laptop Initiative? Perhaps: Tablet PCs and DyKnow Vision Software. Proceedings of Frontiers in Education 2006.

L. D. Merkle, M. C. McClurg, M. G. Ellis, and T. G. Hicks-Wright. EA-Based Generation of Compiler Heuristics for Polymorphous Computing Architectures. Presented in the Military and Security Applications of Evolutionary Computation Workshop of the 2006 Genetic and Evolutionary Computation Conference.

E. Borzello and L. Merkle. Multi-Robot Cooperation Using the Ant Algorithm with Variable Pheromone Placement. Proceedings of the 2005 IEEE Congress on Evolutionary Computation.

R. Layton, J. Holden, T. Hudson, and L. Merkle. Underwater Model Rockets: An Innovative Design Problem and Competition for Undergraduate Students in Engineering, Math and Science. Proceedings of the 2005 American Society for Engineering Education Annual Conference & Exposition. Best paper in Mechanical Engineering Division.

L. D. Merkle and D. J. Burns. Hybrid Architectures for Evolutionary Computing Methods: Automated Transfer of Evolutionary Computation Successes to the Evolvable Hardware Domain. Presented in the Military and Security Applications of Evolutionary Computation Workshop of the 2004 Genetic and Evolutionary Computation Conference.

B. S. Fagin and L. D. Merkle. Measuring the Effectiveness of Robots in Teaching Computer Science. Proceedings of the Thirty-Fourth SIGCSE Technical Symposium on Computer Science Education, 2003 (32% acceptance rate).

A. T. Chamillard and L. D. Merkle. Management Challenges in a Large Introductory Computer Science Course. Proceedings of the Thirty-Third SIGCSE Technical Symposium on Computer Science Education, 2002 (31% acceptance rate).

B. S. Fagin, L. D. Merkle, and T. W. Eggers. Teaching Computer Science Concepts with Robotics Using Ada/Mindstorms 2.0. Proceedings of SIGAda 2001.

L. D. Merkle and J. W. Luginsland. Optimization of the signal growth rate in a class of multicavity RKOs with axially varying geometry using a parallel real-valued evolutionary algorithm. Presented at 2000 IEEE International Conference on Plasma Science.

G. E. Sasser, L. D. Merkle, R. E. Peterkin, et al. Virtual prototyping of RF weapons Challenge Project. Proceedings of the 1999 DoD HPC Users Group Meeting.

L. D. Merkle and J. W. Luginsland. Constrained evolutionary optimization of the signal growth rate in an RKO with an axially varying waveguide diameter. Presented at 1999 IEEE International Conference on Plasma Science.

G. E. Sasser, L. D. Merkle, R. E. Peterkin, et al. Virtual prototyping of microwave devices using MHD, PIC, and CEM codes. Proceedings of the 1998 AIAA Conference.

L. D. Merkle, R. E. Peterkin, et al. Virtual prototyping of RF weapons: A DoD Challenge Project. Proceedings of the 1998 DoD HPC Users Group Meeting.

L. D. Merkle, G. H. Gates, Jr., and G. B. Lamont. Scalability of an MPI-based fast messy genetic algorithm. In Caroll et al., ed., Applied Computing 1998: Proceedings of the 1998 Symposium on Applied Computing, New York: The Association for Computing Machinery, 1998.

L. D. Merkle and G. B. Lamont. A random function based framework for evolutionary algorithms. In Thomas Bäck, editor, Proceedings of the Seventh International Conference on Genetic Algorithms, San Mateo, CA: Morgan Kauffman, 1997 (49% acceptance rate).

C. E. Kaiser, G. B. Lamont, L. D. Merkle, G. H. Gates, Jr., and R. Pachter. Exogenous parameter selection in a real-valued genetic algorithm. In Proceedings of the Fourth IEEE Conference on Evolutionary Computation, Piscataway New Jersey: IEEE Service Center, 1997.

C. E. Kaiser, L. D. Merkle, G. B. Lamont, G. H. Gates, Jr., and R. Pachter. Case studies in protein structure prediction with real-valued genetic algorithms. In M. Heath et al., ed., Proceedings of the Eighth SIAM Conference on Parallel Processing for Scientific Computing, Philadelphia, PA: SIAM, 1997.

C. E. Kaiser, G. B. Lamont, L. D. Merkle, G. H. Gates, Jr., and R. Pachter. Polypeptide structure prediction: Real-valued versus binary hybrid genetic algorithms. In Bryant et al., ed., Applied Computing 1997: Proceedings of the 1997 Symposium on Applied Computing, New York: The Association for Computing Machinery, 1997.

L. D. Merkle, G. H. Gates, Jr., G. B. Lamont, and R. Pachter. Hybrid genetic algorithms for minimization of a polypeptide specific energy model. In Thomas Bäck, Zbigniew Michalewicz, and Hiroaki Kitano, eds., Proceedings of the Third IEEE Conference on Evolutionary Computation, Piscataway New Jersey: IEEE Service Center, 1996.

L. D. Merkle, R. L. Gaulke, G. H. Gates, Jr., G. B. Lamont, and R. Pachter. Hybrid genetic algorithms for polypeptide energy minimization. In Applied Computing 1996: Proceedings of the 1996 Symposium on Applied Computing, New York: The Association for Computing Machinery, 1996.

G. H. Gates, Jr., R. Pachter, L. D. Merkle, and G. B. Lamont. Simple genetic algorithm parameter selection for protein structure prediction. In David Fogel, editor, Proceedings of the Second IEEE Conference on Evolutionary Computation, Piscataway New Jersey: IEEE Service Center, 1996 (70% acceptance rate).

L. D. Merkle and G. B. Lamont. An initial analysis of data parallelism in the fast messy genetic algorithm. In E. Deaton et al., eds., Applied Computing 1994: Proceedings of the 1994 Symposium on Applied Computing, New York: The Association for Computing Machinery, 1994.

L. D. Merkle, and G. B. Lamont. Comparison of parallel messy genetic algorithm data distribution strategies. In S. Forrest, ed., Proceedings of the Fifth International Conference on Genetic Algorithms, San Mateo, CA: Morgan Kaufmann Publishers, Inc., 1993 (34% acceptance rate).

Other

L. D. Merkle. Examples of DyKnow Usage in Computer Architecture I. Presented in Assessing the Impact of Pen-based Computing and Collaboration-facilitating Software on Student Learning, Best Assessment Processes IX Pre-symposium Showcase organized by J. Williams, 2007.

L. D. Merkle. Evolutionary computation in polymorphous computing architectures: Metaoptimization of the Scale in-lining priority function for TRIPS. Final Performance Report, Grant FA8750-05-1-0019, Report Number 07-129-0153, Prepared for Air Force Research Laboratory, 26 Electronic Parkway, Rome, NY 13441-4514, 2007.

L. D. Merkle, M. C. McClurg, and M. G. Ellis. Evolutionary Computation in Polymorphous Computing Architectures. Presented at the DARPA Polymorphous Computing Architectures Principal Investigators' Meeting, March 2006.

L. D. Merkle, M. G. Ellis, and M. C. McClurg. Evolutionary Computation in Polymorphous Computing Architectures. Presented at the DARPA Polymorphous Computing Architectures Principal Investigators' Meeting, August 2005.

L. D. Merkle, M. G. Ellis, Tyler G. Hicks-Wright, and M. C. McClurg. Evolutionary Computation in Polymorphous Computing Architectures. Presented at the DARPA Polymorphous Computing Architectures Principal Investigators' Meeting, March 2005.

L. D. Merkle. Evolutionary Computation in Polymorphous Computing Architectures. Presented to Advanced Computing Architectures Branch of Air Force Research Laboratory, 2005.

L. D. Merkle and K. May. Parallel and Embedded Hardware Implementations of an Evolutionary Computing Optimization Method. Presented to Advanced Computing Architectures Branch of Air Force Research Laboratory, 2004.

L. D. Merkle. Design Optimization for a Novel Class of High Power Microwave Sources Using a Farming Model Parallelization of a Real-Valued Evolutionary Algorithm. Presented at Rose-Hulman Institute of Technology, Parallel Computing Seminar Series, 2003.

A. T. Chamillard and L. D. Merkle. Evolution of an Introductory Computer Science Course: The Long Haul. Presented at the 2002 Rocky Mountain Conference of the Consortium for Computing in Small Colleges.

L. D. Merkle and J. W. Luginsland. Design Optimization for a Novel Class of High Power Microwave Sources: Incorporating Constraints in a Real-Valued Evolutionary Algorithm. Presented at the 2002 Annual Meeting of the American Physical Society Division of Computational Physics.

L. D. Merkle, M. C. Carlisle, J. W. Humphries, and D. W. Lopez. EA-based approach for detecting stealthy attacks. Presented at the 2002 IEEE Workshop on Information Assurance.

L. D. Merkle, J. W. Luginsland. Evolutionary optimization of the signal growth rate in an RKO. Presented at the 1999 High Power Microwave Conference.

L. D. Merkle, J. W. Luginsland. Feasibility of relativistic klystron oscillator design optimization using a real-valued evolutionary algorithm. Presented at the 40th Annual Meeting of the Division of Plasma Physics, 1998.

D. Keefer, M. H. Frese, L. D. Merkle, R. E. Peterkin, Jr., N. F. Roderick, and K. F. Stephens II. Opening mechanisms in an explosively formed fuse opening switch. Presented at the VIIIth Megagauss Conference, 1998.

L. D. Merkle, R. E. Peterkin, Jr., et al. DoD HPC challenge project: Virtual prototyping of RF weapons. Presented at the 8th DoD High Performance Computing Users Group Conference, 1998.

D. Keefer, M. H. Frese, L. D. Merkle, R. E. Peterkin, Jr., N. F. Roderick, and K. F. Stephens II. MACH2 simulation of an Explosively Formed Fuse Opening Switch. Presented at the 1998 IEEE International Conference on Plasma Science, 1998.

R. E. Peterkin, Jr., L. D. Merkle. Two-dimensional simulations of repetitive pulsed laser interactions with solid targets in air. Presented at the 1998 IEEE International Conference on Plasma Science, 1998.

J. J. Watrous, G. E. Sasser, J. W. Luginsland, L. D. Merkle. Three-dimensional particle-in-cell simulations of the relativistic klystron oscillator. Presented at the 1998 IEEE International Conference on Plasma Science, 1998.

L. D. Merkle, R. E. Peterkin, Jr. Three-temperature MHD calculation of the critical surface of laser absorption in laser induced plasmas. Presented at the 39th Annual Meeting of the Division of Plasma Physics, 1997.

L. D. Merkle, G. H. Gates, Jr., and G. B. Lamont. An MPI implementation of the fast messy genetic algorithm. In Proceedings of the Intel Supercomputer Users' Group 1997 Annual North America Users Conference, Beaverton, Oregon: Intel Supercomputer Systems Division, 1997.

C. E. Kaiser, L. D. Merkle, G. B. Lamont, G. H. Gates, Jr., and R. Pachter. Stochastic methods for prediction of modified polypetides. Presented at the First Annual International Conference on Computational Molecular Biology, 1997.

C. E. Kaiser, L. D. Merkle, and G. B. Lamont. Real-valued hybrid genetic algorithms for polypeptide structure prediction. Presented at the 28^{th} Central Regional Meeting of the American Chemical Society, 1996.

G. H. Gates, Jr., R. Pachter, L. D. Merkle, C. E. Kaiser, and G. B. Lamont. Hybrid genetic algorithms and Monte-Carlo with minimization as applied to polypeptide structure determination. Presented at the 28^{th} Central Regional Meeting of the American Chemical Society, 1996.

G. H. Gates, Jr., R. Pachter, L. D. Merkle, and G. B. Lamont. Parallel simple vs. fast messy GAs for protein structure prediction. In Proceedings of the Intel Supercomputer Users' Group 1995 Annual North America Users Conference, Beaverton, Oregon: Intel Supercomputer Systems Division, 1995.

G. H. Gates, Jr., L. D. Merkle, R. Pachter, G. B. Lamont, and W. W. Adams. Polypeptide energy minimization using the parallel fast messy genetic algorithms. Polym. Prepr. 36, 647-648, 1995.

L. D. Merkle, G. H. Gates, Jr., G. B. Lamont, and R. Pachter. Conformational search using a parallel fast messy GA with migration and parallel selection. Presented at the 209th National Meeting of the American Chemical Society, 1995.

L. D. Merkle, G. H. Gates, Jr., G. B. Lamont, and R. Pachter. Application of the parallel fast messy genetic algorithm to the protein folding problem. In Proceedings of the Intel Supercomputer Users' Group 1994 Annual North America Users Conference. Beaverton Oregon: Intel Supercomputer Systems Division, 1994.

G. H. Gunsch, D. E. Dyer, M. J. Gerken, L. D. Merkle, and M. A. Whelan. Autonomous agents as air combat simulation adversaries. In U. Fayyad and R. Uthurusamy, eds., In Proc. SPIE Vol. 1963, p. 50-60, Applications of Artificial Intelligence 1993: Knowledge-Based Systems in Aerospace and Industry, 1993.

D. J. Brinkman, L. D. Merkle, G. B. Lamont, and R. Pachter. Parallel genetic algorithms and their application to the protein folding problem. In J. Wold, ed., Proceedings of the Intel Supercomputer Users' Group 1993 Annual North America Users' Conference, Beaverton Oregon: Intel Supercomputer Systems Division, 1993.

A. Dymek, L. D. Merkle, and G. B. Lamont. Parallelization of standard and messy genetic algorithms. In Proceedings of the Intel Supercomputer Users' Group 1992 Annual Users' Conference, Beaverton Oregon: Intel Supercomputer Systems Division, 1992.

Proposals

In Review

Center for Computational Biology 2007

- Institute invited by Howard Hughes Medical Institute to compete for one time grant of up to $1.6M over 4 years
- Member of six-person committee revising proposal based on direction from Institute President and Vice President for Academic Affairs

Successful

Digilent Development Boards for CSSE 232 Computer Architecture I 2006

- Five Digilent Spartan Starter 3E Boards

"Evolutionary Computation in Polymorphous Computing Architectures" 2004-2007

- One time $64K grant from Air Force Research Laboratory

"Advanced Computing Technology Branch Evolvable Hardware Support" 2004

- One time $13K grant from Air Force Research Laboratory

Xilinx Software and Digilent Development Boards 2003

- One time donation valued at $225K from Xilinx University Program for Computer Architecture I
- 90 licenses for full version of Xilinx ISE Foundations 5.2 (replaced $40 per copy student edition)
- Five Digilent Digilab D2E Development Boards (allowed students to implement project designs on state-of-the-art FPGAs)

"Information Warfare and Network Security" 2000-2002

- Renewable $50K per year grant from National Security Agency

"Advanced Evolutionary Computing for Directed Energy Applications" 1999-2002

- DoD High Performance Computing Modernization Program project

"Computational Magnetohydrodynamics" 1999

- Coordinator for proposal to renew Common High Performance Computing Software Support Initiative project

"Virtual Prototyping of RF Weapons" 1998

- Coordinator for proposal to renew DoD Challenge Project

Others

"Leading the Way in a New Field: Bioinformatics Courses and an Interdisciplinary Bioinformatics Program," Co-PI for RHIT "Success Grant" $98K grant over two years 2005

"Using a Common Suite of Integrated EDA Tools throughout an Electrical and Computer Engineering Curriculum to Improve Student Learning of Engineering," Proposal Team Member 2003, 2004

"Using GQM for Program, Curriculum, and Course Assessment," Co-PI for $170K grant over 36 months from NSF 2003

"CAREER: Advanced Evolutionary Algorithms Theory and Techniques for Computational Science and Engineering Applications," one time $463K grant over 60 months from NSF CAREER Grant proposal 2003

Society Memberships

Air Force Association (Life Member)
American Association for the Advancement of Science
American Society of Engineering Education
Association for the Advancement of Artificial Intelligence
Association of Computing Machinery. Special Interest Groups (varies):

- Ada (SIGADA)
- Algorithms and Computation Theory (SIGACT)

- Applied Computing (SIGAPP)
- Artificial Intelligence (SIGART)
- Computer Science Education (SIGCSE)
- Evolutionary Computation (SIGEVO)

Institute of Electrical and Electronics Engineers. Societies (varies):

- Computational Intelligence (formerly Neural Networks)
- Computer
- Education
- Nuclear and Plasma Sciences
- Systems, Man, and Cybernetics

Conferences, Workshops, and Reviews

General Assembly (Disciples of Christ)	Fort Worth, TX 2007
Genetic and Evolutionary Computation Conference	London, England, UK 2007
Innovative Technologies in Computer Science Education	Dundee, Scotland, UK 2007
Best Assessment Processes Symposium IX	Terre Haute, IN 2007
Regional Assembly (Disciples of Christ, Indiana Region)	Indianapolis, IN 2006
IEEE Congress on Evolutionary Computation	Vancouver, BC, Canada 2006
Genetic and Evolutionary Computation Conference	Seattle, WA 2006
Undersea Defence Technology: Europe	Hamburg, Germany 2006
IEEE SMC Information Assurance Workshop	West Point, NY 2006
Best Assessment Processes Symposium VIII	Terre Haute, IN 2006
ACM Special Interest Group on Computer Science Education	Houston, TX 2006
ABET Faculty Assessment Workshop Version 2.0	San Diego, CA 2005
Frontiers In Education	Indianapolis, IN 2005
IEEE Congress on Evolutionary Computation	Edinburgh, UK 2005
General Assembly (Disciples of Christ)	Portland, OR 2005
DoD/NASA Evolvable Hardware Workshop	Washington, DC 2005
Genetic and Evolutionary Computation Conference	Washington, DC 2005
IEEE SMC Information Assurance Workshop	West Point, NY 2005
ACM Special Interest Group on Computer Science Education	St. Louis, MO 2005
Evolutionary Computation in Polymorphous Computing Architectures	
• Kickoff Meeting	Rome, NY 2005
• PCA Principal Investigators Meeting	Scottsdale, AZ 2005
• PCA Principal Investigators Meeting	Boulder, CO 2005
Regional Assembly (Disciples of Christ, Indiana Region)	French Lick, IN 2004
DARPA Polymorphous Computing Architectures PI Meeting	Monterey, CA 2004
Genetic and Evolutionary Computation Conference	Seattle, WA 2004
DoD/NASA Evolvable Hardware Workshop	Seattle, WA 2004
IEEE Congress on Evolutionary Computation	Portland, OR 2004
Information Institute General Workshop	Rome, NY 2004
IEEE SMC Information Assurance Workshop	West Point, NY 2004
Colloquium on Information Systems Security Education	West Point, NY 2004
Microsoft TechEd 2004	San Diego, CA 2004
Best Assessment Processes Symposium VI	Terre Haute, IN 2004
ACM Special Interest Group on Computer Science Education	Norfolk, VA 2004
IEEE Congress on Evolutionary Computation	Canberra, Australia 2003
General Assembly (Disciples of Christ)	Charlotte, VA 2003
Pi Kappa Alpha Officers Leadership Academy	Memphis, TN 2003
GECCO Afterglow Workshop	Champaign-Urbana, IL 2003
Genetic and Evolutionary Computation Conference	Chicago, IL 2003

IEEE SMC Information Assurance Workshop	West Point, NY 2003
DoD High Performance Computing Users Group Meeting	Seattle, WA 2003
High Performance Computing Advisory Panel Meeting	USAF Academy, CO 2003
Best Assessment Processes Symposium V	Terre Haute, IN 2003
ACM Special Interest Group on Computer Science Education	Reno, NV 2003
Women in Information Technology	Indianapolis, IN 2002
Annual Meeting of the Division of Computational Physics	San Diego, CA 2002
Genetic and Evolutionary Computation Conference	New York, NY 2002
IEEE SMC Information Assurance Workshop	West Point, NY 2002
ACM Special Interest Group on Computer Science Education	Covington, KY 2002
NSA review of Network Security and Information Warfare	Ft. Meade, MD 2001
Genetic and Evolutionary Computation Conference	San Francisco, CA 2001
IEEE International Conference on Plasma Science	Las Vegas, NV 2001
IEEE SMC Information Assurance Workshop	West Point, NY 2001
ACM Special Interest Group on Computer Science Education	Charlotte, NC 2001
ABET Open Enrollment Faculty Workshop	San Juan, PR 2001
Genetic and Evolutionary Computation Conference	Las Vegas, NV 2000
IEEE International Conference on Plasma Science	New Orleans, LA 2000
Congress on Evolutionary Computation	Washington, D.C.1999
High Power Microwave Conference	Albuquerque, NM 1999
IEEE International Conference on Plasma Science	Monterrey, CA 1999
DoD High Performance Computing Users Group Meeting	Monterrey, CA 1999
ACM Symposium on Applied Computing	San Antonio, TX 1999
Annual Meeting of the Division of Plasma Physics	New Orleans, LA 1998
DoD High Performance Computing Modernization Office review of Computational Electromagnetics and Acoustics Computational Technology Area	WPAFB, OH 1998
DoD High Performance Computing Users Group Meeting	Houston, TX 1998
International Conference on Evolutionary Computation	Anchorage, AK 1998
DoD High Performance Computing Modernization Office Common High-Performance Computing Software Support Initiative Alpha Test and review of Computational Magnetohydronamics	WPAFB, OH 1998
Workshop on Parallel Profiling and Debugging	Vicksburg, MS 1998
ACM Symposium on Applied Computing	Atlanta, GA 1998
Annual Meeting of the Division of Plasma Physics	Pittsburgh, PA 1997
International Conference on Genetic Algorithms	East Lansing, MI 1997
Intel Supercomputing Users Group Meeting	Albuquerque, NM 1997
SIAM Conf. on Parallel Processing for Scientific Computing	Minneapolis, MN 1997
American Chemical Society Central Regional Meeting	Dayton, OH 1996
International Conference on Evolutionary Computation	Perth, Australia 1995
U. Illinois Workshop: Fast Messy Genetic Algorithms	Champaign-Urbana, IL 1995
International Conference on Genetic Algorithms	Pittsburgh, PA 1995
Intel Supercomputing Users Group Meeting	Albuquerque, NM 1995
AFOSR Workshop on Optimization of Molecular Structures	Washington, D.C.1995
National Meeting of the American Chemical Society	Los Angeles, CA 1995
Intelligent Systems for Molecular Biology	Palo Alto, CA 1994
International Conference on Evolutionary Computation	Orlando, FL 1994

International Conference on Genetic Algorithms — Champaign-Urbana, IL 1993
AFOSR Workshop: Optimization Techniques for Large Compounds — Ames, IA 1993

Intel Supercomputing Users Group Meeting — Dallas, TX 1992
International Conference on Parallel Processing — St. Charles, IL 1992

World Congress on Expert Systems — Orlando, FL 1991
Oak Ridge National Laboratory Tenth Parallel Circus — Oak Ridge, TN 1991

Training and Self Improvement

Personal Productivity System based on Getting Things Done: The Art of Stress Free Productivity, Dave Allen, and Take Back Your Life: Using Outlook to Get Organized and Stay Organized, Sally McGhee — 2007
DoD Information Assurance Awareness Training — 2007

Elders' Pre-Assembly Conference — 2005, 2007
Structural Bioinformatics: A BioQUEST Curriculum Consortium Approach — 2005

Not-for-profit Board Development Workshop — 2004
Rose-Hulman Fall Writeoff — 2004
Course audit: Compiler Construction, Prof. Claude Anderson — 2004
Course audit: Computer Architecture II, Prof. Tina Hudson — 2004
Course audit: Computer Security, Prof. Mark Ardis — 2004
Evolutionary Bioinformatics: A BioQUEST Curriculum Consortium Approach — 2004
Air Command and Staff College — 1999-2004

Information Warfare Applications Course — 2003
Guidant, Inc. visit focusing on computer architectures for pacemakers — 2003

Reading Enhancement Course — 2001
Course Assessment Seminar — 2001
Center for Educational Excellence Seminars

- "Preparing a Portfolio for Professional Growth and Promotion" — 2001
- "Active Learning and Cooperative Groups in the Lecture Classroom" — 2000
- "Interacting with Front Page" — 2000

Academy Character Enrichment Seminar — 1999
USAFA New Instructor Orientation — 1999

Laser Short Course — 1998

Acquisition Fundamentals — 1997

Squadron Officers School — 1994

Introduction to Acquisition Management — 1990

Air Force Logistics Command Materiel Management — 1988

SERVICE

Students

Rose-Hulman Institute of Technology

Tau Beta Pi sponsored Fundamentals of Engineering Exam Review Session (scheduled) — 2008

New Student Orientation, Social, Professional, and Ethical Expectations — 2002-Present
- Computing Perspective in Introductory Session (2006,2007)
- Small Group Session Leader (2002)

Faculty Advisor, Tau Beta Pi Honor Society — 2005-Present

Rose-Hulman Chorus — 2003-Present
Faculty Co-advisor (2004-Present)

Faculty Advisor, Programming Contest Teams — 2003-Present
- ACM Intercollegiate Programming Contest (2003 – Present)
- Carnegie Mellon University Invitational Programming Contest (2005, 2007)

Faculty Advisor, Upsilon Pi Epsilon Honor Society — 2003-Present

Academic Advisor — 2003-Present
- 20 CS and SE majors (2007-2008)
- 14 Freshmen and four upperclass SE majors (2006-2007)
- 23 CS and SE majors (2003-2006)

Participant, Mobile Computing Study — 2007-2008

Judge, CSSE Laboratory Design Contest — 2007

Driver, Lambda Chi Alpha Run for Kids' Sake — 2007

Chapter Advisor, Pi Kappa Alpha Social Fraternity (Iota Delta chapter) — 2003-2007

Thesis Committee Member:
- M.S. Thesis, Justin Dillman — 2005-Present
- M.S. Thesis, Doug Morgan — 2003-Present
- M.S. Thesis, Curtis A. Schmitt — 2003-Present
- M.S. Thesis, Harsha V. Yarlagadda — 2003-2005

Midwest Undergraduate Private Engineering Colleges Design Competition — 2004

Client, Term Project, *Software Architecture* — 2004

Laptop Orientation — 2003, 2004

Client, Term Project, *Software Requirements and Specification* — 2003

Client, Procedure Project, *Technical Communication* — 2002

United States Air Force Academy

Faculty Advisor, ACM Student Chapter 2001-2002

Associate Air Officer Commanding, Cadet Squadron 21 2001-2002

Cadet Summer Research Program Representative 2000-2002
- Arrange summer positions with outside organizations (Air Force, DoD, and other government organizations)
- Manage logistical issues associated with cadet travel and performance evaluation

Sponsor Family, six USAF Academy cadets 2000-2002

Academic Advising 1999-2002
- Computer Engineering Assistant Advisor-In-Charge (2000-2002)
- Advisor for over 50 cadets, including cadets majoring in computer science, computer engineering, and basic sciences, as well as undeclared cadets (1999-2002)

Associate Air Officer Commanding, Basic Cadet Training B Squadron 2001

Officer Member, Cadet Wing Honor Board 2001

Shadow Program 2001

Air Force Institute of Technology

Thesis Committee Member:
- Ph.D. Thesis, Jesse Zydallis, Air Force Institute of Technology, *Building-Block-Based Multiobjective Messy Genetic Algorithms: Theory, Analysis, and New Innovations*. 2001-2003
- M.S. Thesis, David Caswell, Air Force Institute of Technology, *Active Processor Scheduling Using Evolutionary Algorithms*. 2002
- M.S. Thesis, Steve Michaud, Air Force Institute of Technology, *Solving the Protein Structure Prediction Problem with Fast Messy Genetic Algorithms*. 2001
- Ph.D. Thesis, David Van Veldhuizen, *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*, Air Force Institute of Technology. 1999
- M.S. Thesis, Karl Deerman, *Protein Structure Prediction Using Parallel Linkage Investigating Genetic Algorithms*, Air Force Institute of Technology. 1999

Department

Rose-Hulman Institute of Technology, Computer Science and Software Engineering

Computer Science Program Vision Statement Committee 2007-2008

Computer Science Program Coordinator 2004-Present
- Lead implementation of CSSE Continuous Course Improvement Process (2004-Present)
- ABET Computing Accreditation Commission Self-Study (2005-2006) – program accredited

- Led development of CSSE Continuous Course Improvement Process (2004-2005)

Operating Systems and Computer Security Lab Equipment Committee 2004-Present

Fundamentals of Software Development Committee 2003-2006

Awards and Honors Ceremony 2004

Chair, Fundamentals of Software Development Committee 2003-2004

New Faculty Mentor, Prof. Archana Chidanandan 2003-2004

Honors and Awards Committee 2003

United States Air Force Academy, Computer Science

Director of Core Instruction 2001-2002

- Overall responsibility for introductory computing course
  - Graduation requirement for all cadets
  - Supervised 22 instructors teaching 59 sections
  - 1171 cadets completed course
  - Represents over half of department's teaching workload
- Responsible for textbook selection, development of course materials, development and maintenance of course website, and recommendation of course grades to the Dean
- Ensure consistent grading and dissemination of information about graded events by all instructors

Led department-wide redesign of introductory computing course 2000-2002

- Added application-level learning objectives for algorithms, systems, databases, and other non-programming topics to make course relevant to cadets in non-computing majors
- Added web-based pre-assessment quizzes covering reading at the knowledge and comprehension learning levels, allowing class time to focus on more difficult application level objectives.
- Emphasized active and collaborative teaching techniques
- Incorporated classroom use of standard issue laptops
- Simplified programming syntax to allow instructors to focus on principles underlying programming constructs
- Instituted Honors version for advanced cadets – covers same topics in more depth, as well as additional programming topics
- Outstanding results:
  - Students better prepared for class
  - Higher grades, supporting observation of better learning
  - Despite the course's bad reputation, students rated it highest of any core course in Basic Sciences or Engineering in 9 of 36 categories, including "amount learned" and "course as a whole"

Supervisor 2000-2002

- Civilian faculty (one associate professor)
- Military faculty (one assistant professor and three instructors, all captains)
- Newly commissioned lieutenant awaiting pilot training, performing some duties similar to a graduate assistant

Computer Science Curriculum Committee 1999-2002

Deputy for Computer Engineering 2001

- Department focal point for issues associated with new Computer Engineering major, jointly administered with Department of Electrical Engineering
- Serve as Division Head in course assessment process for all Computer Science courses taken by Computer Engineering majors – assist course directors in development of Course Assessment Plans, review Course Assessment Reports, develop and deliver Division Assessment Report

Research Director, USAFA Department of Computer Science 2000-2001

- Hired Research Associate
    - Faculty are heavily loaded with teaching duties, and do not have graduate students to pursue interesting areas of research
    - Department has several sources of external funding
- Disseminated information about research opportunities (implemented web page and database to organize information)
- Maintained records of department research
    - Prepare annual Department Research Review/Summary
    - Assisted in self-study for CSAB accreditation.
    - Prepare department submissions for both internal and external research bulletins
- Reviewed department research proposals and publications
- Coordinated Independent Study courses
- Point of contact for Air Command and Staff College research topic “Information Operations”

Institution

Rose-Hulman Institute of Technology

Chair, Advisory Committee on Academic Computing 2007-Present

Beta Tester, Automated Absence Notification System 2006-Present

Parallel Computing Steering Committee 2005-Present

Greek Advisory Council 2003-Present

Math Advisory Committee 2003-Present

Focus Group Member, Academic Governance Commission 2007

Howard Hughes Medical Institute Undergraduate Science Education Proposal Team 2007

ABET “Supergroup” 2005-2007

Faculty Affairs Committee 2005-2007

Laptop Computer Committee 2005-2006

Rules and Discipline Committee 2004-2005

Second Year Faculty Perspective, New Faculty Dinners 2003-2004

Secretary, Quality of Education Committee 2003-2004

Judge, MATHCOUNTS 2003

Visual and Performing Arts Committee 2002-2003

United States Air Force Academy

Process Improvement Principle, Department of Electrical Engineering 2001-2002

Co-Chair, Computer Engineering Curriculum Committee 1999-2002

USAFA Engineering Criteria 2000 Committee 1999-2002

- Represent Department of Computer Science in the development of assessment plan for computer engineering program
- Initial ABET visit for computer engineering program in Fall of 2002 led to accreditation

Chair, Computer Engineering Working Group 2000-2001

- Under the guidance of the Computer Engineering Steering Group, coordinate the joint administration of the computer engineering major between the Department of Computer Science and the Department of Electrical Engineering

Officer of the Day (about once per semester) 2001

Basic Sciences Division representative for selection of Thomas D. Moore Award winner for outstanding research in the Cadet Summer Research Program 2000

Summer Scientific Seminar coordinator 2000

- Each summer, several hundred prospective applicants attend seminars offered by the academic departments during each of two weeks. One of USAFA's most effective recruiting tools.
- Coordinated department's offering of "Programming for the World Wide Web," in which students use web authoring tools to develop a home page
- Student feedback was extremely positive

Chapman University

Represented Department of Computer Science in negotiating articulation agreement with College of Santa Fe. 1998

Profession

Genetic and Evolutionary Computation Conference 1999, 2003-Present

- Program Committee, Genetic Algorithms Track (1999, 2004-2006)
- Undergraduate Workshop
    - Chair/co-chair (2004-Present)
    - Panel Member (2003-Present)

- Workshop on Defense Applications of Computational Intelligence (formerly Workshop on Military and Security Applications of Evolutionary Computation)
    - Co-chair (2004-Present)

Web-based SIGCSE Conference Registration System – Lead designer, developer, and maintainer 2004-Present

Technical Symposium on Computer Science Education 2003-Present
- Co-registrar (2003-Present)
- Judge, Doctoral Consortium (2003)

Innovative Technologies in Computer Science Education 2007-Present
- Registrar and Treasurer (2007-2008)
- Reviewer (2007-Present)

WWW@10 Conference, Host of Distinguished Guest 2004

IEEE International Conference on Systems, Man, and Cybernetics 2002-2004
- Reviewer (2003-2004)
- Reviewer, Student Paper Competition (2003)
- Tutorials Chair (2002-2003)

United States Air Force Scientific Advisory Board Study, Technical Editor 2000
- *Science & Technology and the Air Force Vision: Achieving a More Effective S&T Program*

HPM Generation Seminar, Initiator and coordinator 1998-1999

Intel Supercomputer Users Group Meeting, Vendor Coordinator 1997

External Reviewer:
- Panel Member, DoE Accelerated Strategic Computing Initiative PSE/DisCom2 Milepost Review (2001)
- Proposal Reviewer, Common High Performance Computing Software Support Initiative Computational Electromagnetics and Acoustics project (1999)
- Beta Test reviewer, DoD Common High Performance Computing Software Support Initiative project EIGER – Electromagnetic Interactions GeneRalized (1999)
- Proposal Reviewer, Department of Energy Small Business Innovative Research (1998-1999)

Journal Article Referee:
- Annals of Operations Research (2007)
- IEEE Transactions on Systems, Man, and Cybernetics (2000, 2001, 2007)
- Journal of Interactive Learning Environments (2006, Sp2007, Fa2007)
- Inverse Problems in Engineering (2006)
- Genetic Programming and Evolvable Machines special issue on Biological Applications of Genetic and Evolutionary Computation (2003)

- Genetic Programming and Evolvable Machines special issue on Computation in Gene Expression (2001)
- Evolutionary Computation special issue on Scalable Evolutionary Computation (two articles) (1999)

Conference Paper Referee:

- IEEE SMC Information Assurance Workshop (4 papers, 2002)
- Intel Supercomputer Users Group Meeting (3 papers/year, 1994-1997)
- Parallel Problem Solving from Nature (5 papers, 2002)
- Symposium on Applied Computing (4 papers/year, 1994-Present)

Session Chair:

- Frontiers in Education (2005)
- Congress on Evolutionary Computation (2004)
- ACM Symposium on Applied Computing (1998)
- Intel Supercomputer Users Group Conference (1997)

Community

Volunteer, Lost Creek Elementary School Computer Laboratory 2007-2008

Assistant Cubmaster, Cub Scout Pack 200 2007-2008

Rose-Hulman United Way Campaign Representative 2007

Wildwood Day Camp Den Walker 2007

Tiger Den Leader, Cub Scout Pack 200 2006-2007

Substitute Teacher, World Gospel Church Homeschool Algebra Class 2005

Judge, Community Theatre of Terre Haute 2003-2004

Rose-Hulman Daycare Committee 2002

Judge, Mountain Ridge Middle School Science Fair 2001

Judge, New Mexico High School Supercomputing Challenge 1997-1999

Volunteer, Habitat for Humanity 1992, 1997

Judge, Miami Valley Regional Science Fair 1993-1996

Assistant Scoutmaster, Troop 85 1984-1987

Religious

Indiana Commission on United Ministries in Higher Education

Board of Directors 2004-Present

- Chair, Review and Consultation Committee (2005-Present)
- Personnel Committee (2004-2005)

Chair, Review and Consultation Team, Fort Wayne Campus Ministry 2005

United Campus Ministry of Terre Haute, IN

Board of Directors 2003-Present

- Chair (2006-2007)
- Representative to Indiana Commission of United Ministries in Higher Education (2004-Present)
- Executive Committee (2004-2007)
- Finance Committee (2004-Present)
- Vice Chair (2005)
- Chair, Building and Grounds Committee (2004-2006)

Central Christian Church of Terre Haute, IN

Board of Elders 2005-Present

- Chair (2005)
- Representative to General Board (2007-Present)

Congregational Representative, *Disciples World* (monthly denominational magazine) 2003-Present

“New Frontiers” Sunday School class leader (occasional) 2003-2005

Regional Assembly Delegate 2004, 2006

General Assembly Delegate 2003, 2005, 2007

Moderator (two terms) 2004-2005

- Chair of General Board
- Major rewrite of Constitution & By-Laws
- Successful Senior Minister search
- Development and adoption of vision statement
- Successful debt retirement Capital Campaign
- Ex-officio member of all church committees

Chair, Vision Statement Committee 2004

Interim Minister Search Committee 2003

Member, Chancel Choir 2002-2003

Member, Men’s Choir 2002-2003

Member, Worship Committee 2003

First Christian Church of Colorado Springs, CO

Chair, Capital Campaign Task Force 2001-2002
Deacon 2001-2002
Member, Chancel Choir 1999-2002

Monte Vista Christian Church of Albuquerque, NM

Co-chair, Membership Ministry 1998-1999
Deacon 1997-1999
Member, Chancel Choir 1997-1999

Central Christian Church of Kettering, OH

Member, Nominating Committee	1996
Chair, Debt Retirement Campaign	1996
President, Chancel Choir	1995-1996
Secretary, Bridge Club	1995-1996
Member, Stewardship Committee	1995-1996
Member, Chancel Choir	1992-1996
Deacon	1990-1996
Co-leader, Christian Children's Fellowship	1994-1995
"Jeffrey" in the musical Godspell	1992
Worship Leader	1990

HONORS

Academic

Who's Who in Engineering Education	2005-2006
Best Paper, Mechanical Engineering Division, 2005 American Society fo r Engineering Education Annual Conference & Exposition	2005
USAFA Department of Computer Science Research Excellence Award	2001-2002
Upsilon Pi Epsilon	2001
Winning team of Service Academy Faculty Programming Contest	2000-2001
Technical Editor for United States Air Force Scientific Advisory Board	2000
Eta Kappa Nu	1992
Tau Beta Pi	1992
National Merit Semifinalist	1983
Presidential Scholar	1983

Leadership and Service

Rose-Hulman Institute of Technology Faculty Member of the Week (5 times)	2003-2005
Credentialed Space Professional	2004
Outstanding Associate Air Officer Commanding, 2nd Basic Cadet Training	2001
Air Force Achievement Medal	2000
Air Force Meritorious Service Medal	1999
Outstanding Briefing Letter, Squadron Officer School	1994
Air Force Commendation Medal	1991
Air Force Reserve Officer Training Corps Scholarship	1983-1987
Eagle Scout	1984

REFERENCES

Col David S. “Hoot” Gibson, Ph.D.
Permanent Professor and Head of Computer Science
HQ USAFA/DFCS
2354 Fairchild Dr.
USAF Academy, CO 80840-6234
719-333-3590
david.gibson@usafa.af.mil

Gary B. Lamont, Ph.D.
Professor of Computer Science
AFIT/ENG
4950 P. Street
Wright-Patterson AFB, OH 45433
937-255-3636 x4718
lamont@afit.af.mil

John W. Luginsland, Ph.D.
Nuclear Engineer
NumerEx
700 Warren Rd #14-3A
Ithaca, NY 14850
607-277-4272
john.luginsland@numerex.com

David C. Mutchler, Ph.D.
Professor of Computer Science and Software Engineering
Rose-Hulman Institute of Technology
5500 Wabash Ave., CM 99
Terre Haute, IN 47803
812-877-8426
mutchler@rose-hulman.edu