Jeffrey D. Oldham

144 Wyandotte Dr, San Jose, CA 95123-3727
408.578.3806
oldham@cs.stanford.edu
http://theory.stanford.edu/~oldham/

Objective

Software Development and Algorithmic Research

Experience and Skills

Software Engineer: CodeSourcery, LLC (June 2000–). Design, specify, and implement the C++ VSIPL++ toolkit supporting vector, signal, and image processing in both serial and distributed environments. Complete USAF OSD Phase 1 SBIR to implement VSIPL++ extensions to support multiple processors and SIMD instructions. Write *Advanced Linux Programming* and POOMA user manual. Specify and develop QMTest test regression tool. Write run-time C++ library. Optimize financial programs. Perform patent law work. Compiler development including writing C++ ABI test library, optimization work, bug fixes.

Assistant Professor: Trinity University Computer Science Department (August 1999–May 2000). Teach courses in introductory programming (two semesters), operating systems, and simulation. With 44 students, design and implement sales tax server prototype. With 12 students, develop models and simulate investment accumulation and contraction.

Visiting Assistant Professor: Dartmouth College Computer Science Department (Spring 1999). Taught advanced algorithms.

AT&T Labs Researcher: AT&T Research Labs (Summer 1998). The proposed HTTP 1.1 protocol permits persistent connections between servers and clients. Design and evaluate policies to predict when to terminate persistent server-client connections.

Stanford University Ph.D. Candidate: Computer Science Department (September 1992–August 1999). Design theoretically-fast algorithms for multicommodity and generalized flow problems; multicommodity flow implementation was several orders of magnitude faster than previous implementations. Design and experimentally evaluate faster algorithms to value Asian options. Implement DNA sequence alignment algorithm, used in MIS 214 course. Instructor for compiler course and programming languages courses. Teaching assistant for intensive C++ and compiler optimization courses.

Recent Software, Specifications, and Publications

VSIPL++: Specify and implement the C++ Vector, Signal, and Image Processing Library for use in both serial and distributed environments using intuitive, algorithmic syntax. Write 250+ page specification incorporating the best current software research. Work funded by the Department of Defense High Performance Embedded Computing Software Initiative coordinated through MIT Lincoln Laboratory (http://www.hpec-si.org/).

Multicommodity Flow: Combinatorial approximation program was faster than its closest competitor by several orders of magnitude.

***Advanced Linux Programming*:** Book describing UNIX and Linux-specific library programming interfaces. Joint work with Mark Mitchell and Alex Samuel.

POOMA: Develop and maintain open-source C++ software toolkit for parallel scientific computation. Users include nuclear weapon physicists at Los Alamos National Laboratory. http://www.codesourcery.com/pooma/pooma/.

POOMA User's Manual: User manual for the open-source POOMA C++ toolkit for parallel scientific computation. http://www.codesourcery.com/public/pooma/manual.pdf.

Education

Ph.D. in Computer Science: Stanford University, August 1999. Thesis title: "Multicommodity and Generalized Flows: Theory and Practice." Development and implementation of algorithms to solve multicommodity and generalized flow problems. In the first half, I converted theoretically fast but, in practice, slow algorithms into an algorithm that was several orders of magnitude faster than its closest competition. In the second half, I developed the first combinatorial algorithms to solve many generalized flow problems.

M.S. in Computer Science: Stanford University, June 1996.

B.A. in Computer Science: Rice University, May 1991. Summa cum laude. National Merit Scholar. Tau Beta Pi.