

Jeffrey D. Oldham

938 Clark Ave. #34, Mountain View, CA 94040-2244

650.968.0708

oldham@cs.stanford.edu

<http://theory.stanford.edu/~oldham/>

Objective

Software Development and Algorithmic Research

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.

Experience and Skills

Software Engineer: CodeSourcery, LLC (June 2000–). Compiler development including optimization work and scientific code development.

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, designed and implemented sales tax server prototype. With 12 students, developed models and simulated 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 clients. Designed and evaluated policies to predict when to terminate persistent server-client connections.

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

Research Programmer: Center for Research on Parallel Computation (May 1991–May 1992). Added color graphics to user interface of tool to extract parallelism from Fortran programs.

Chairperson of Administrative Council: First United Methodist Church, Sunnyvale, CA (1996–99). Oversaw administration, church programming, and goal setting; initiated writing of employee policies and procedures manual

Programming: Development of programs optimized for fast execution, development of programs implementing theoretical results, programming for hypothesis testing; C++, C, Scheme, Perl, UNIX, shell, CPLEX, Mathematica, \LaTeX .

Recent Software

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

Generalized Flow: generalized shortest path program asymptotically faster than linear programming

Sales Tax Server: server software prototype to determine sales tax for any location and any set of items; under development as Trinity class project with forty-four students

Technical Illustrations: programmed most illustrations for Donald E. Knuth’s *The Art of Computer Programming*

DNA Sequence Alignment: used in Stanford’s MIS 214 course

Retirement Income Simulation: simple model determining withdrawal rates; for Trinity University homework exercise