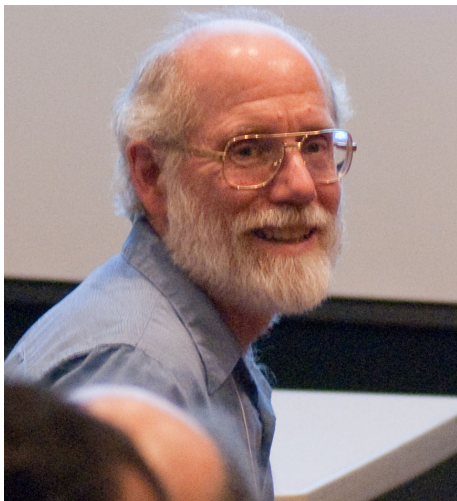


Análise Experimental de Algoritmos: as contribuições de David S. Johnson

Luciana Buriol	Instituto de Informática, UFRGS
Celina Figueiredo	Engenharia de Sistemas e Computação, UFRJ
Eduardo Uchoa	Departamento de Engenharia de Produção, UFF



http://dimacs.rutgers.edu/DIMACS_highlights/djohnson/djohnson.html

David S. Johnson
1945–2016

Análise Teórica de Algoritmos

Disciplina que se firmou no final dos anos 60 e nos anos 70. Alguns marcos:

- ▶ Publicação de “The Art of Computer Programming” de Donald Knuth (1968)
- ▶ Tese de Cobham-Edmonds (algoritmo bom = algoritmo polinomial)
- ▶ Teoria da NP-completude (Cook-Karp)
- ▶ Outros nomes fundamentais: Rabin, Aho, Hopcroft, Ullman, DSJ, Tarjan, Papadimitriou, ...

Ênfase no comportamento assintótico e na análise do pior caso.

Análise Teórica de Algoritmos: Limitações

Exemplo notório de que nem sempre o pior caso conta a história toda:

- ▶ O algoritmo simplex para PL é exponencial (Klee e Minty, 1972), mas tem excelente desempenho prático.

Análise probabilística: tenta descrever o comportamento esperado de um algoritmo

- ▶ Assume hipóteses discutíveis sobre qual seria a distribuição dos dados nas “instâncias típicas”
- ▶ Mesmo com as distribuições mais simples (uniforme, normal, etc), é difícil de realizar para algoritmos complexos

Análise Teórica de Algoritmos: Limitações

Como analisar algoritmos para problemas NP-difíceis?

Algoritmos exatos

- ▶ A análise de pior caso pouco diz, todos os algoritmos conhecidos são exponenciais

A principal alternativa sugerida em GJ79 para lidar com esses problemas era usar **algoritmos aproximados**. Entretanto:

- ▶ Muitos problemas NP-difíceis importantes não podem ser bem aproximados
- ▶ Por focar no pior caso, muitos algoritmos aproximados funcionam mal em instâncias típicas, bem pior do que **heurísticas** sem nenhuma garantia.

Análise Experimental de Algoritmos (AEA)

Nas palavras de DSJ:

“(AEA) address questions of determining realistic algorithm performance where worst case analysis is overly pessimistic and probabilistic models are too unrealistic. (...) experimentation can provide guides to realistic algorithm performance where analysis fails.”

Análise Experimental de Algoritmos (AEA)

A partir dos anos 80, DSJ passou a ter um forte interesse por AEA e fez grandes contribuições:

- ▶ Publicou vários trabalhos exemplares. Exemplos: as análises da metaheurística Simulated Annealing e de heurísticas para o problema do caixeiro-viajante
- ▶ Escreveu extensamente sobre metodologia, buscando estabelecer padrões rigorosos para AEA
- ▶ Liderou a criação do SODA (ACM-SIAM Symposium on Discrete Algorithms)
- ▶ Criou e foi o principal organizador dos DIMACS Challenges

DIMACS Implementation Challenges

Center for Discrete Mathematics and Theoretical Computer Science (DIMACS):

- ▶ Colaboração entre as universidades de Rutgers e Princeton e laboratórios AT&T, Bell Labs, Applied Communication Sciences e NEC.
- ▶ Fundado em 1989, conta com 250 membros permanentes.

Desde 1990 o DIMACS apoia (com pessoal técnico e recursos computacionais) a realização dos Challenges.

DIMACS Implementation Challenges

Cada Challenge é dedicado a um grupo específico de problemas e tipicamente envolve as seguintes etapas:

1. Definição das instâncias de benchmark, algumas de aplicações reais outras de geradores aleatórios, mas sempre diversificadas e representativas
2. Apresentação do ambiente computacional e das regras da competição
3. Submissão dos códigos participantes, acompanhados de textos descritivos
4. Workshop incluindo apresentações e divulgação dos resultados
5. Publicações com artigos dos participantes e/ou análises comparativas gerais feitas pelos organizadores

Cada Challenge costuma resultar num salto quantitativo e qualitativo na pesquisa de um problema

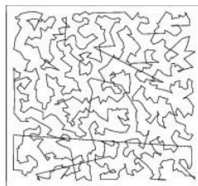
Os 11 DIMACS Challenges

- ▶ 1991: Network Flows and Matching
- ▶ 1993: Maximum Clique, Graph Coloring, and Satisfiability
- ▶ 1994: Effective Parallel Algorithms for Combinatorial Problems
- ▶ 1995: Fragment Assembly and Genome Rearrangements
- ▶ 1996: Priority Queues, Dictionaries, and Multi-Dimensional Point Sets
- ▶ 1999: Near Neighbor Searches
- ▶ 2000: Semidefinite and Related Optimization Problems
- ▶ 2001: The Traveling Salesman Problem
- ▶ 2006: The Shortest Path Problem
- ▶ 2012: Graph Partitioning and Graph Clustering
- ▶ 2014: Steiner Tree Problems

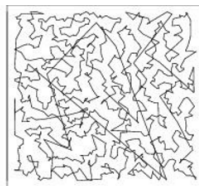
8th DIMACS Implementation Challenge:

The Traveling Salesman Problem

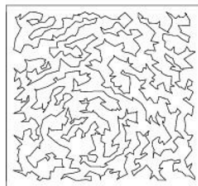
Página inicial do
Challenge sobre
o problema do
caixeiro-viajante
(2001)



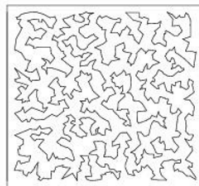
Greedy Tour



Nearest Neighbor Tour



Savings Tour

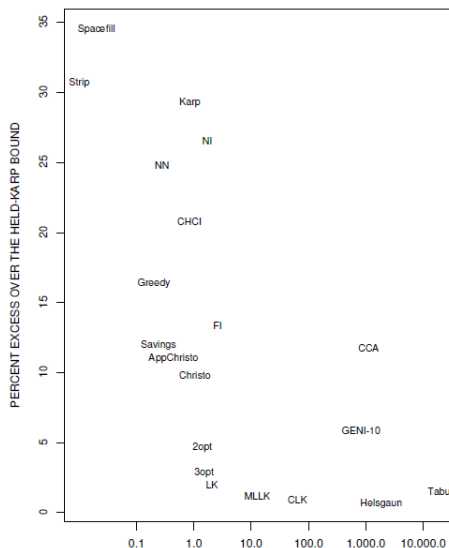


Optimal Tour

Challenge News: Still Open for Business!

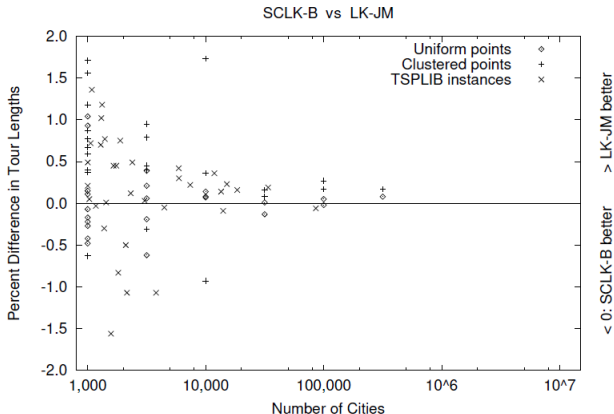
Experimental Analysis of Heuristics for the STSP, 2002 (com L. McGeoch)

Comparação de diversas heurísticas (qualidade da solução \times tempo) em instâncias aleatórias de 10 mil cidades



Experimental Analysis of Heuristics for the STSP, 2002 (com L. McGeoch)

Comparação
instancia a
instância entre
duas variantes
da heurística de
Lin-Kernighan



A Theoretician's Guide to the Experimental Analysis of Algorithms (2002)

Texto metodológico amadurecido ao longo de uma década, DSJ discute sistematicamente toda a área de AEA. Organizado em torno de 10 princípios, são feitas várias sugestões e identificadas as armadilhas (erros mais comuns).

Os 10 princípios:

1. Faça experimentos dignos de nota
 - ▶ Isto é; investigue questões de real interesse da comunidade
 - ▶ Exemplo de armadilha: investigar algoritmos ultrapassados

A Theoretician's Guide to the Experimental Analysis of Algorithms (2002)

2. Relacione seu artigo com a literatura
3. Faça testes em um conjunto de instâncias que permita se chegar a conclusões gerais
 - ▶ Uma questão fundamental em AEA: como obter/gerar bons benchmarks
4. Projete os experimentos de forma eficiente e efetiva
 - ▶ Use técnicas (exemplo: redução de variância) para que conclusões sólidas possa ser obtidas com uma quantidade menor de testes.

A Theoretician's Guide to the Experimental Analysis of Algorithms (2002)

5. Use implementações suficientemente eficientes

- ▶ Às vezes uma implementação eficiente exige estruturas de dados e/ou técnicas sofisticadas
- ▶ Nesses casos, infelizmente, uma implementação ingênua pode invalidar os resultados do estudo
- ▶ “Pet Peeve” (= algo que irrita particularmente uma pessoa): artigos que mencionam conhecimento limitado de programação (ou pior, falta de tempo!) como desculpa

A Theoretician's Guide to the Experimental Analysis of Algorithms (2002)

6. Garanta a reprodutibilidade
7. Garanta a comparabilidade
8. Reporte a história completa
9. Chegue a conclusões bem-justificadas
 - ▶ O trabalho experimental deve fornecer mais do que dados
10. Apresente os dados de modo informativo
 - ▶ Discussão sobre como usar tabelas, figuras e gráficos

Algumas notas sobre a figura humana de David S. Johnson

Conhecido por ser modesto, generoso e um bom líder

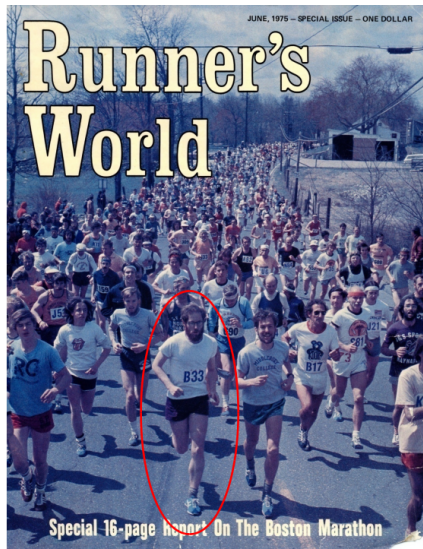
- ▶ Gostava de interagir e encorajar os estudantes. Respondia até mensagens de estudantes desconhecidos outros países
- ▶ Convidava diferentes grupos para almoçar todos os dias na AT&T
- ▶ Organizava o churrasco anual do grupo de teoria do AT&T/Bell-Labs
- ▶ Tentava assistir a todos os seminários que podia

David S. Johnson era um perfeccionista em vários aspectos

- ▶ Sempre fazia minuciosas revisões de artigos, dando inúmeras sugestões de melhorias. A ponto de várias vezes autores oferecerem co-autoria ao revisor anônimo
- ▶ Aceitou fazer revisões de vários manuscritos (alguns bastante amadores) que afirmavam ter resolvido a questão $P=NP$, sempre tentando explicar claramente ao autor o ponto errado na prova
- ▶ Gastava muito tempo tentando aprimorar seus próprios escritos e estava sempre aberto a ouvir sugestões
- ▶ Participou de todas as edições do SODA e de quase todas as edições do STOC e do FOCS ao longo da sua carreira

Mais sobre David S. Johnson

- ▶ Foi casado duas vezes
- ▶ Era um corredor de maratonas, preferia uma alimentação leve e saudável
- ▶ Tinha uma coleção de 5000 CDs, gostava muito de Bossa Nova
- ▶ Sempre insistia para que sua inicial do meio “S.” fosse incluída no seu nome
- ▶ Tinha número de Erdos 2
- ▶ Era a favor de revisões duplo-cegas



Picture taken from David's presentation on his Knuth Prize

Thanks David!
You are a champion!