Autotuning under Tight Budget Constraints: A Transparent Design of Experiments Approach

Autotuning: Optimizing Program Configurations

- How to write efficient code for each of these?
- We can use autotuning:
  - The process of automatically finding a configuration of a program that optimizes an objective

Strategies for Exploring Search Spaces

System	Domain	Approach
ATLAS	Dense Linear Algebra	Exhaustive
INSIEME	Compiler	Genetic Algorithm
Active Harmony	Runtime	Nelder-Mead
ParamILS	Domain-Agnostic	Stochastic Local Search
OPAL	Domain-Agnostic	Direct Search
OpenTuner	Domain-Agnostic	Ensemble
MILEPOST GCC	Compiler	Machine Learning
Apollo	GPU kernels	Decision Trees

Exhaustive, Meta-Heuristics, Machine Learning

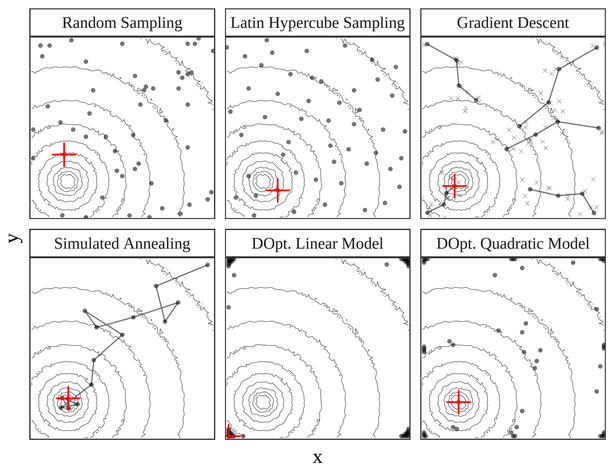
- These approaches need a large number of function evaluations, assuming seach space "smoothness", and that good solutions are reachable
- After optimizing, we learn "nothing" about the search space, and can't explain why optimizations work

Design of Experiments: Exploration under a Budget

Run	A	B	C	D	E	F	G
1	1	-1	1	-1	-1	1	1
2	1	1	1	-1	1	-1	-1
3	-1	1	-1	-1	1	1	1
4	-1	1	1	1	-1	1	-1
5	1	-1	-1	1	1	1	-1
6	1	1	-1	1	-1	-1	1
7	-1	-1	1	1	1	-1	1
8	-1	-1	-1	-1	-1	-1	-1

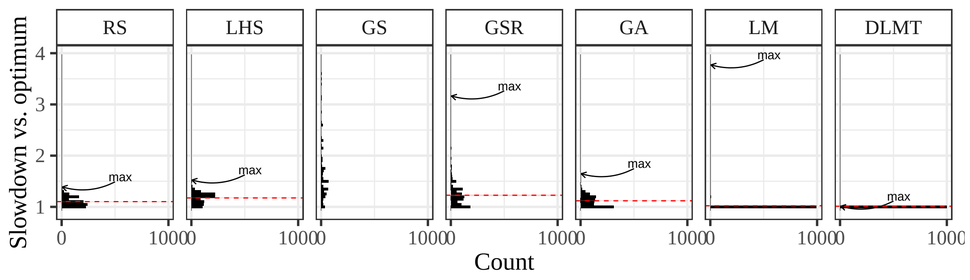
A Plackett-Burman design for 7 2-level factors

- Experiment results can be used to identify relevant parameters and build a performance model

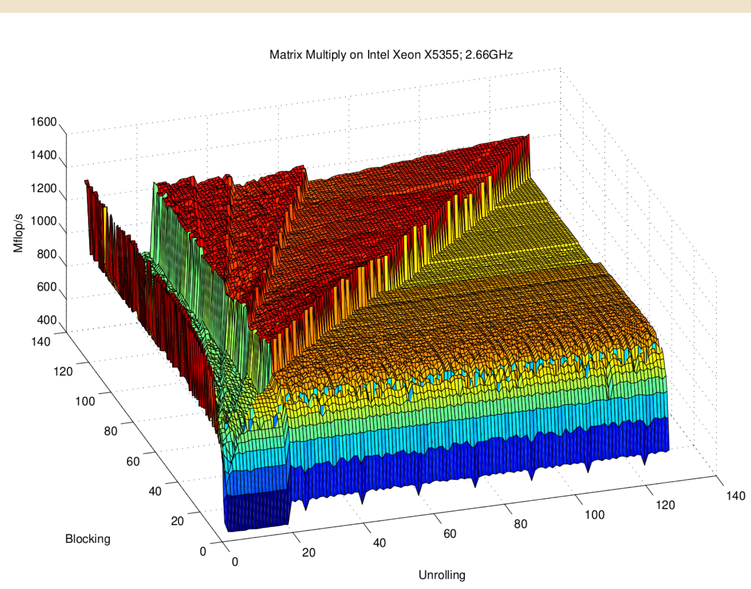


- Exploration of a search space using a fixed budget of 50 points, the red "+" represents the best point found by each strategy

A Motivating Result on a GPU Kernel



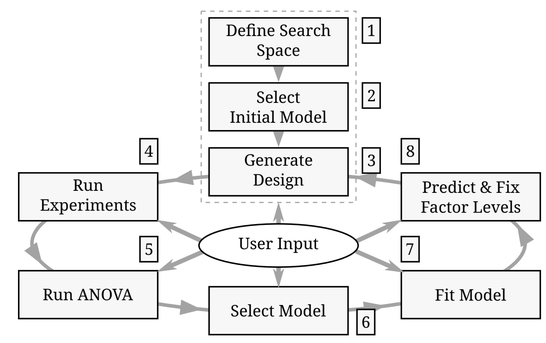
Autotuning: Search Spaces are Hard to Explore



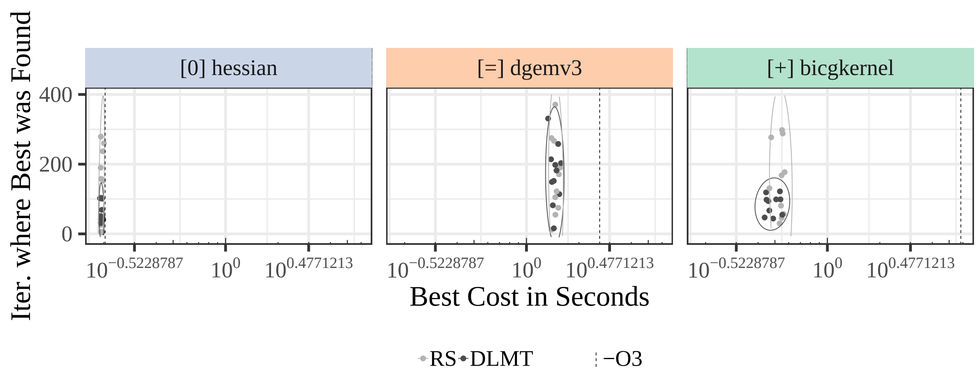
Unrolling, blocking and Mflops/s for matrix multiplication
Seymour K, You H, Dongarra J. A comparison of search heuristics for empirical code optimization. InCLUSTER 2008 Oct 1 (pp. 421-429)

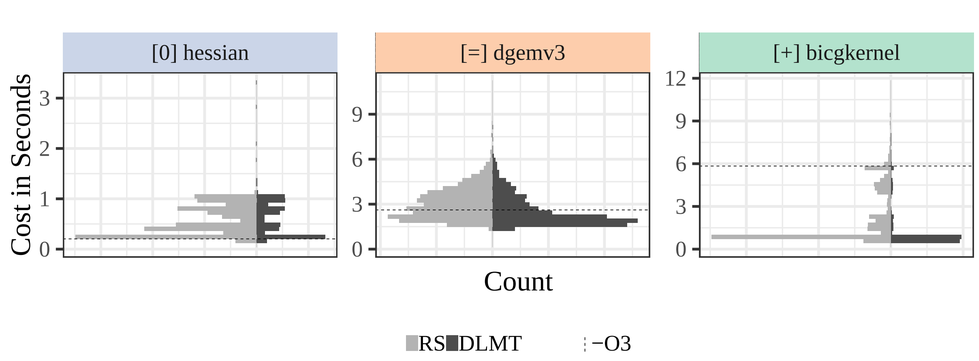
- Represent the effect of all possible configurations on the objectives, can be difficult to explore, with multiple local optima and undefined regions
- Main issues are exponential growth, geometry, & measurement time

Our Design of Experiments Approach



Extensive Evaluation on the SPAPT Benchmark





Pedro Bruel - phrb@ime.usp.br