Optimizing Moolloy
A Solver for Multi-Objective Optimization Problems

TEAM AMALGAM

JOSEPH HONG, CHRIS KLEYNHANS, MING-HO YEE, ATULAN ZAMAN

The Value Packaging Problem

Bob

Wendy

https://flic.kr/p/e3tTRA

https://flic.kr/p/bteiEi

https://flic.kr/p/bPgU6n

Single-Objective Optimization?

Compute a weighted sum.

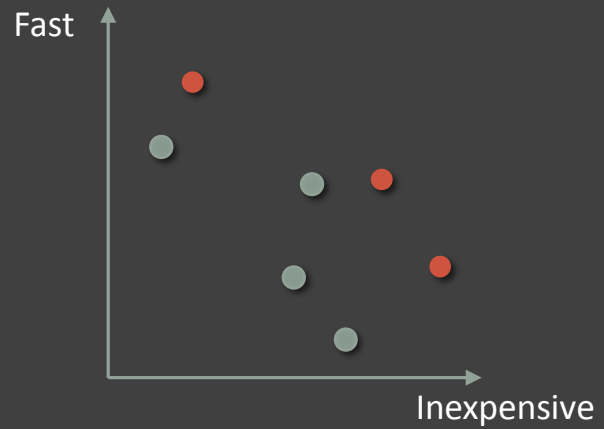
Solve a single-objective optimization problem.

$$\sum w_i x_i$$

But we can do better.

Multi-Objective Optimization

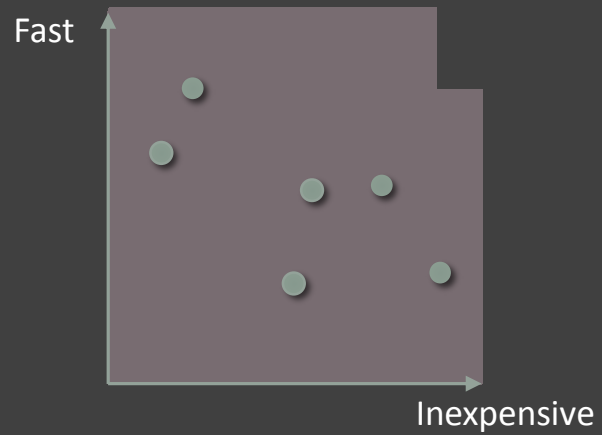
Pareto optimal solutions:



Exact not approximate, **discrete** not continuous

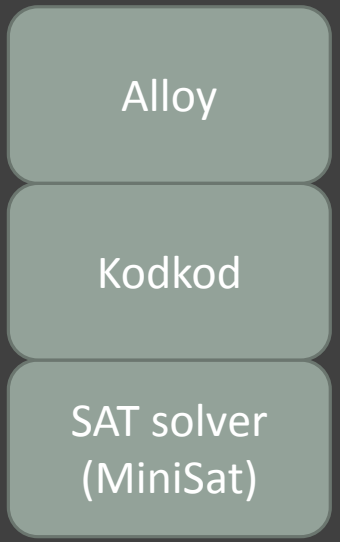
Guided Improvement Algorithm (GIA)

Find all Pareto optimal solutions.



Areas for improvement: speed and scalability

Moolloy System Architecture



Compiles model for Kodkod

GIA implementation

Constraint solver

Two Approaches

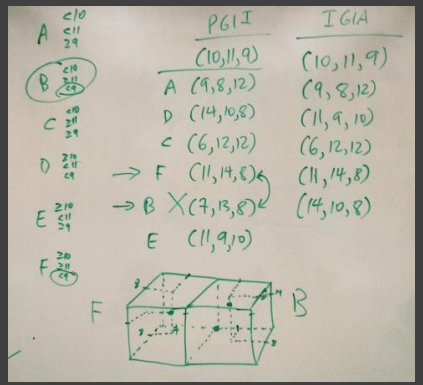
Engineer a better tool

Checkpointing + formula rewriting

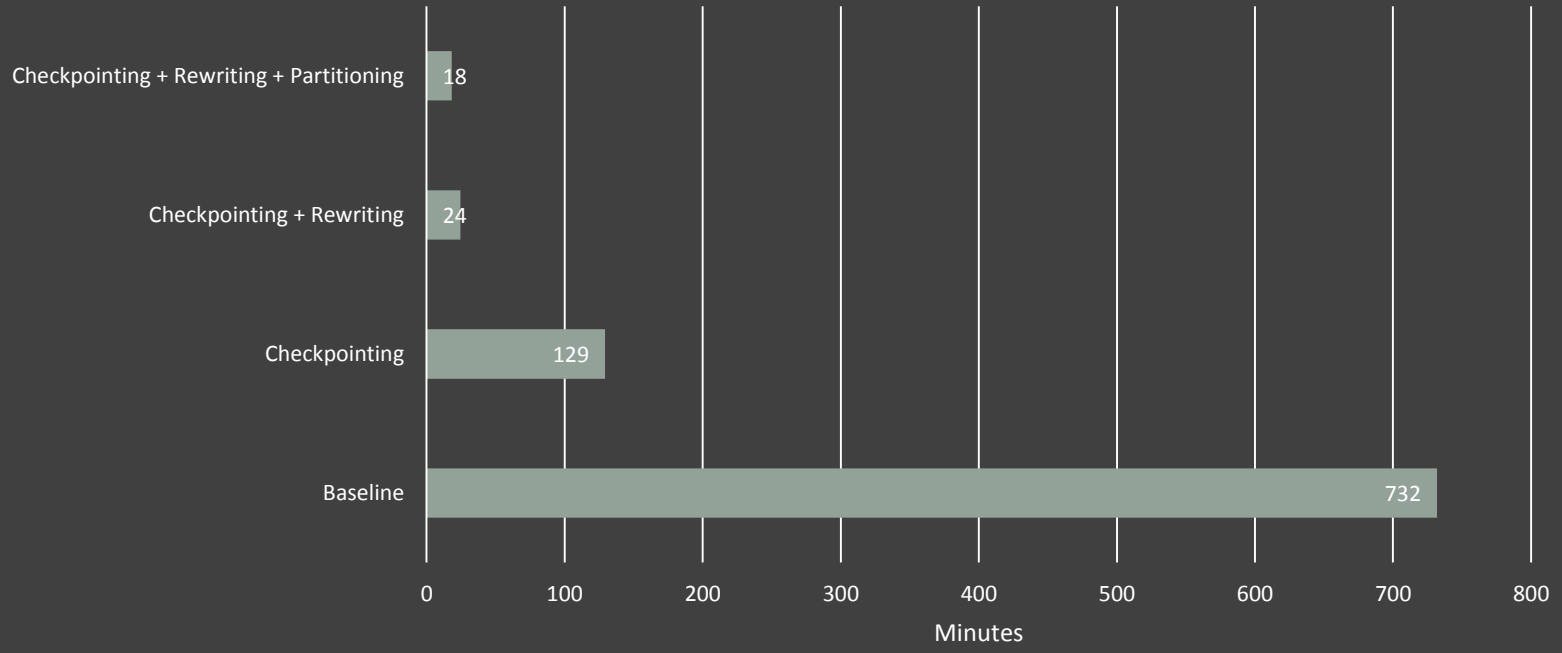
https://flic.kr/p/5rCjjx

Design a better algorithm

Partitioning for parallelism

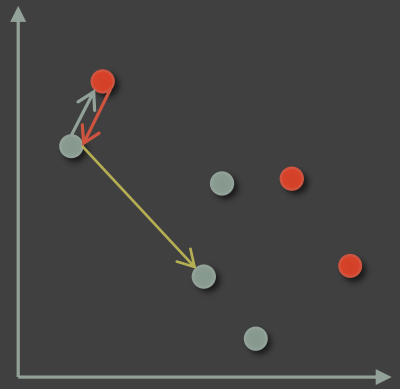


Value Packaging Solve Time



Checkpointing

GIA involves stepping up and backtracking.



We added functionality to save and reuse state.

Formula Rewriting

By rewriting formulas, we can eliminate variables.

Before:

```
(total_cost == electrical + plumbing)
  AND
(total_cost < 100)
```

After:

```
(electrical + plumbing < 100)
```

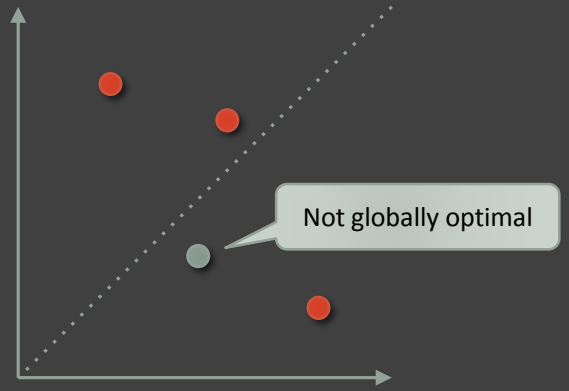
Partitioned GIA (PGIA)

How can we multi-thread the algorithm?

https://flic.kr/p/9AscDz

Splitting the Search Space

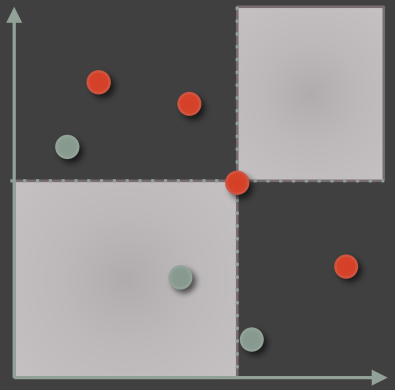
A *locally optimal* solution should be *globally optimal*.



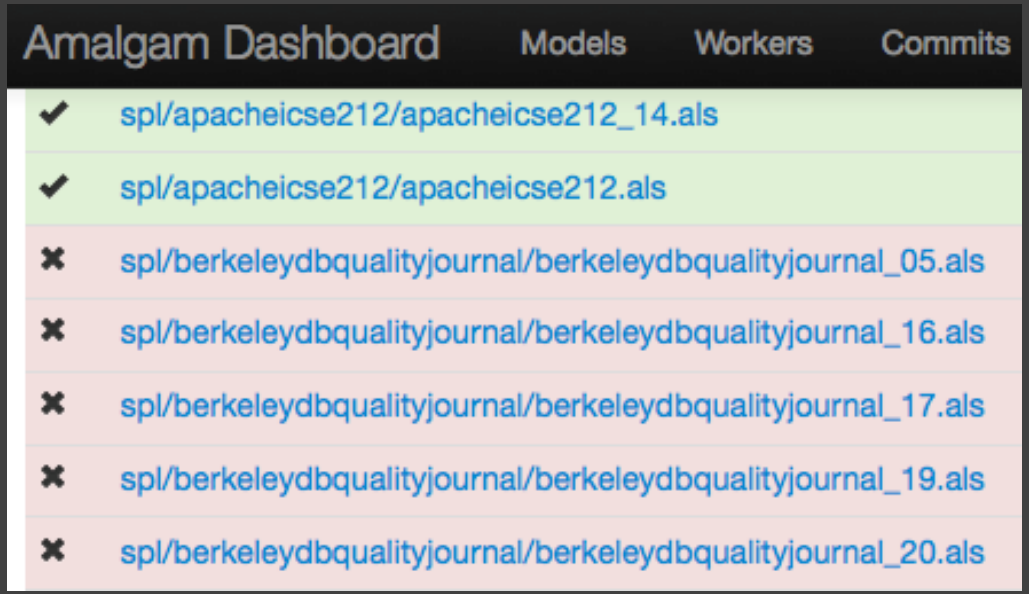
Can we guarantee locally optimal = globally optimal?

Locally Optimal = Globally Optimal

Find a Pareto point, then split the search space.

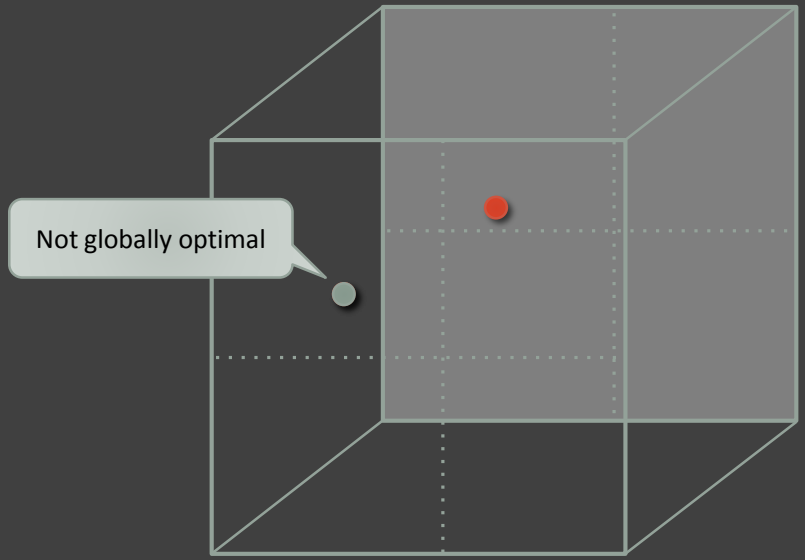


Whoops…

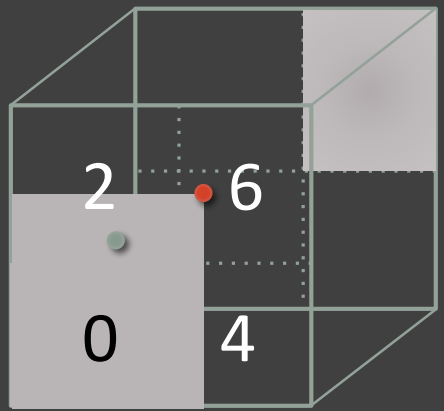


"Beware: Ideas that seem to intuitively work in two dimensions do not always generalize to three or more dimensions."

Locally Optimal ≠ Globally Optimal

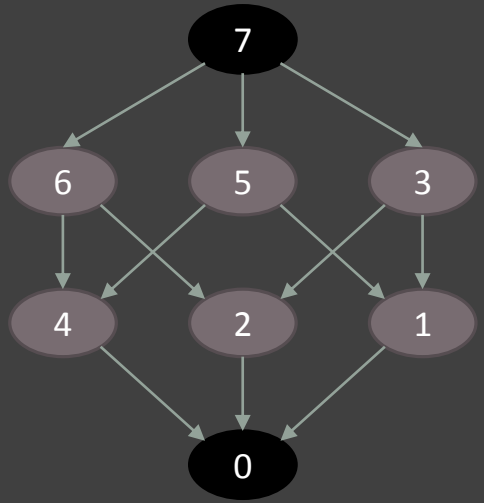


Search Order Matters



3 7

1 5

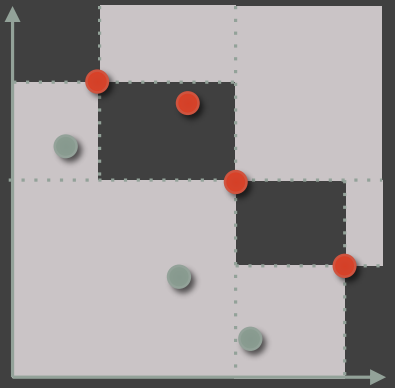


Future Work

Improve engineering

Improve algorithms

More case studies



Run PGIA recursively

Conclusions

Checkpointing + rewriting + partitioning
Average 200x speedup

Paper accepted by ABZ ’14

Value packaging problem solved in
18 minutes (originally: 12 hours)

We’re preparing a paper