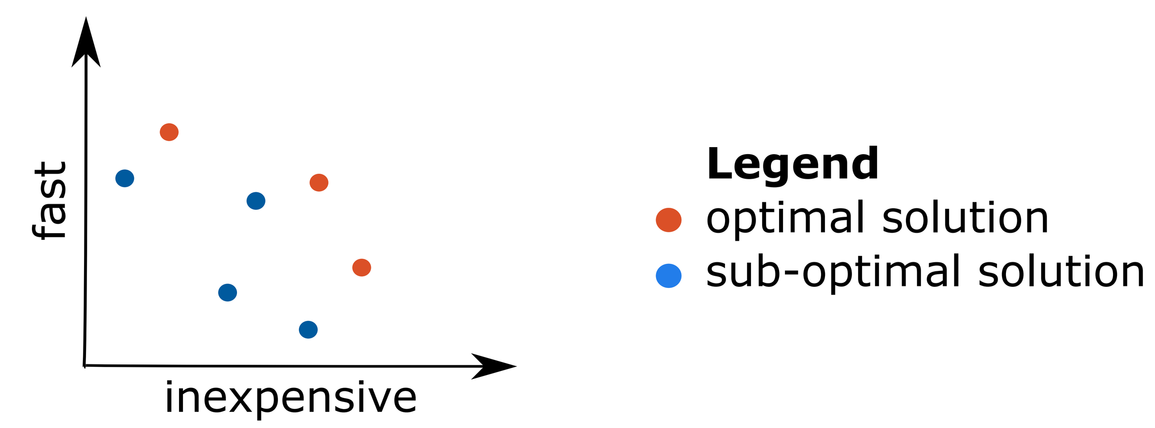
Capstone: INNOVATIVE by Design®

Optimizing Moolloy

A Solver for Multi-Objective Optimization Problems

Problem Description

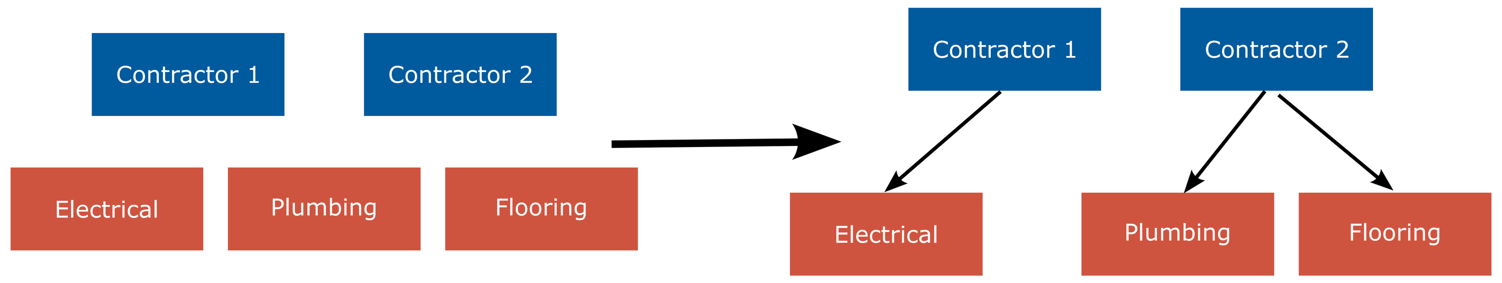
Our project is to optimize Moolloy, a solver for multi-objective optimization problems, so that it can solve problems faster and scale with additional hardware.



Moolloy finds all the optimal solutions, which are configurations that cannot be improved in one metric without making some other metric worse.

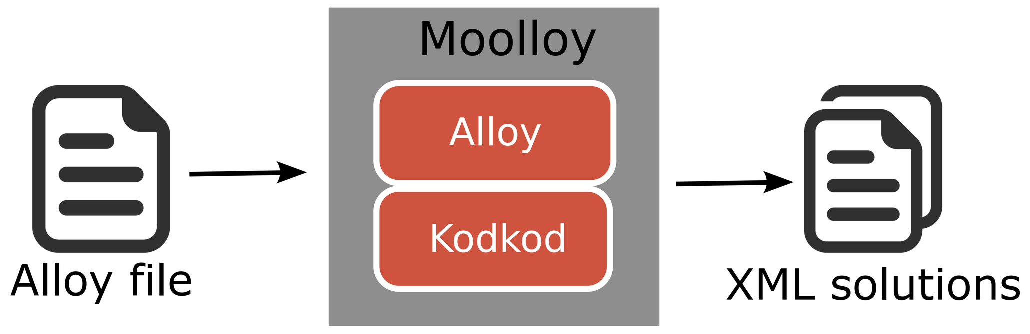
Case Study

In construction projects, work is divided into units which must then be assigned to contractors. The value packaging problem is to find the optimal assignments, based on criteria such as cost, time and experience.



System Architecture

Moolloy is built upon Alloy and Kodkod. Problems are provided in modified Alloy syntax, converted to first order logic with relations, and then solved by Kodkod. Finally, solutions are returned as XML files by Alloy.



Checkpointing

The algorithm solves for an initial solution and solves for better solutions by adding constraints. The last solution found is optimal. To find other optimal solutions, the algorithm removes constraints and repeats the process from a new starting point.

With checkpointing, we save the results of intermediate computations. This improves performance as we can reuse the results when searching for new solutions.

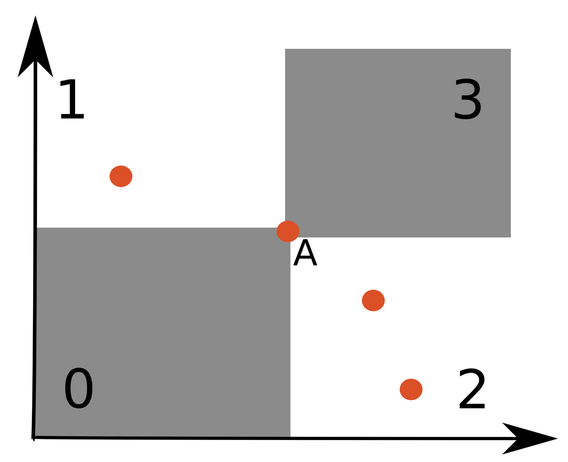
Formula Rewriting

We eliminate variables by rewriting formulas. This simplifies the first order logic formulas, making them easier to solve and also reduces memory usage.

```
(cost == electrical + plumbing) AND (cost < 100)
↓
(electrical + plumbing < 100)
```

Partitioning for Parallelism

We can run the algorithm in parallel to take advantage of multi-core processors, thus improving performance. This is done by partitioning the search space in such a way that locally optimal solutions are globally optimal. This means tasks can find solutions independently without any duplicate work.



Quadrants 1 and 2 can be searched independently for optimal solutions.

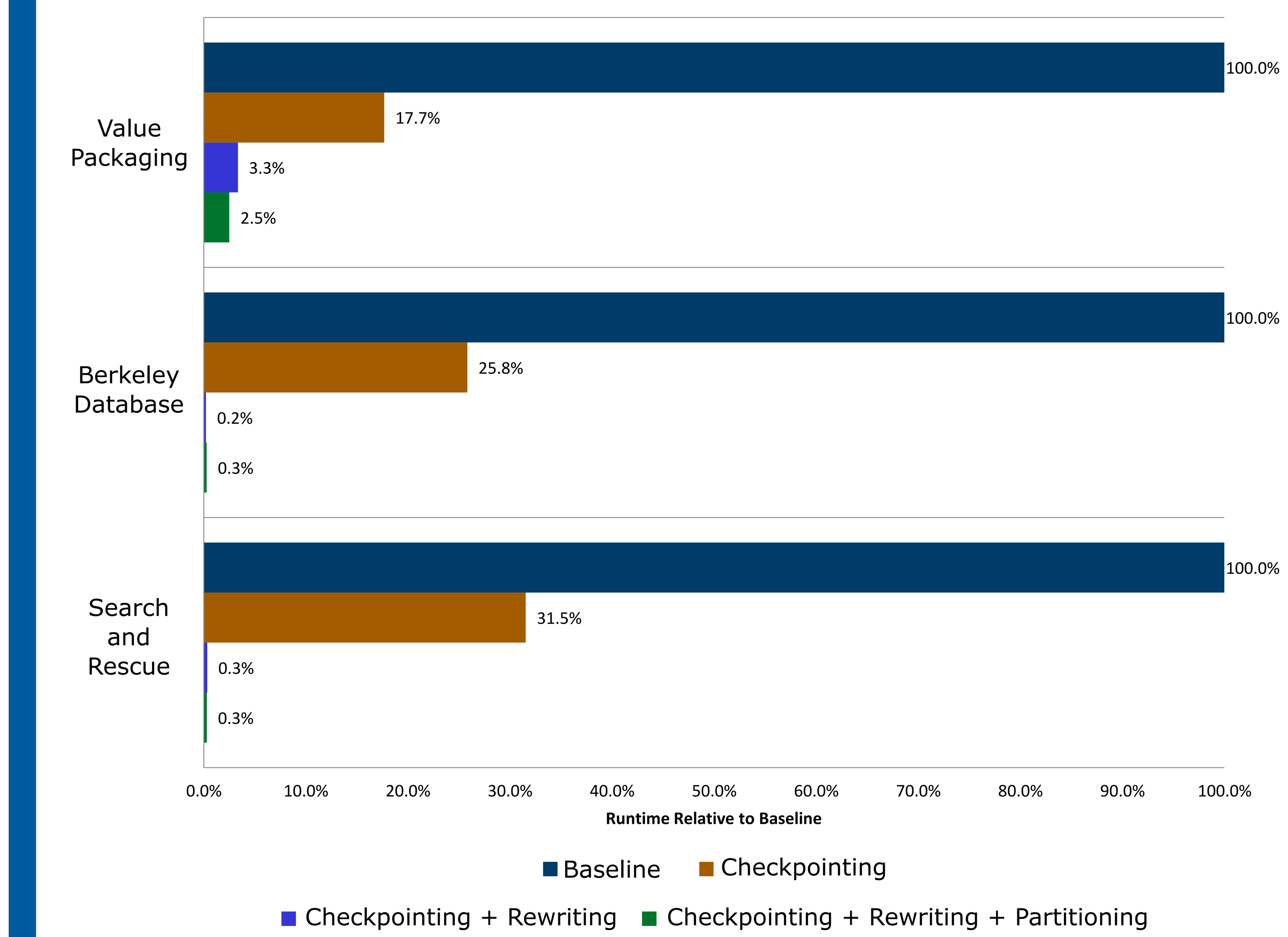
Quadrant 0 is excluded because all solutions are sub-optimal to point A.

Quandrant 3 is excluded because there are no solutions. Otherwise point A would be sub-optimal.

Case Study Results

To demonstrate the benefits of our improvements, we ran benchmarks on the value packaging problem, as well as case studies from software engineering.

This value packaging problem has nine contractors, ten work units, and five metrics. The software engineering problems are feature models with three metrics for Berkeley Database and five metrics for the Search and Rescue system.



Team Amalgam

Joseph Hong, Chris Kleynhans, Ming-Ho Yee, Atulan Zaman

Customer: Derek Rayside

Acknowledgements: Krzysztof Czarnecki, Jianmei Guo, Rafael Olaechea, Mahdi Safa, Ed Zulkoski

For more information, see: mhyee.com/fydp.html

Sponsored by yelp

UNIVERSITY OF WATERLOO