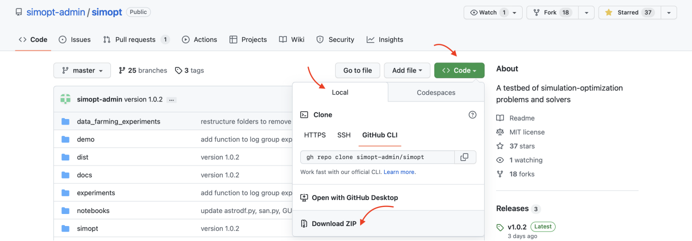
Please follow the steps below before we get started...

1. In your browser, navigate to `https://github.com/simopt-admin/simopt`.
2. Click on "Download ZIP" as shown.



3. Unzip the folder `simopt-master` and open it in VS Code using "File > Open Folder".

SimOpt

2022 Winter Simulation Conference

David J. Eckman, Texas A&M University
Shane G. Henderson, Cornell University
Sara Shashaani, North Carolina State University

Goals and plan

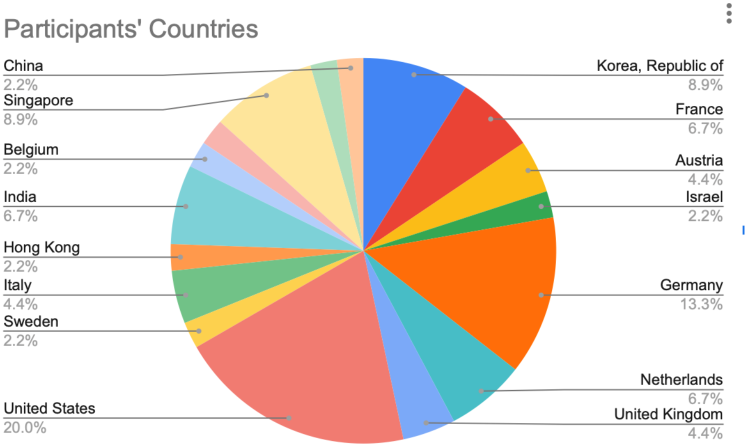
Our goal is for you to learn how to:

- Run SimOpt experiments
- Understand the plots
- Use the GUI
- Build and contribute your own models, problems and solvers

The plan:

1. Comparing solvers with cool plots
2. Using the SimOpt GUI
3. Writing your own models, problems and solvers

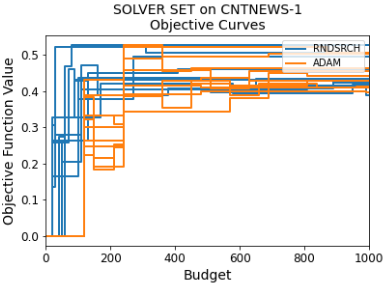
Today's participants



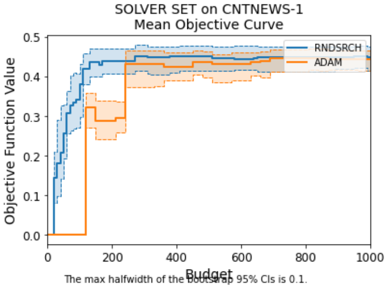
Newsvendor problem

- Per-unit order cost $c = 5$, selling price $s = 9$, salvage value $w = 1$
- Real-valued order quantities
- Demand $D \sim F(x) = 1 - (1 + x^{\alpha})^{-\beta}$ where $x \geq 0, \alpha = 2, \beta = 20$ (this is Burr Type XII)
- Maximize expected profit, where we use simulation to estimate expected profit rather than explicit formulas
- Which of the following two solvers is better?
  1. ADAM
  2. Random search (exp(1)), with 10 reps per solution
- Budget of 1000 replications, start at $x_0 = 0$

Progress curves



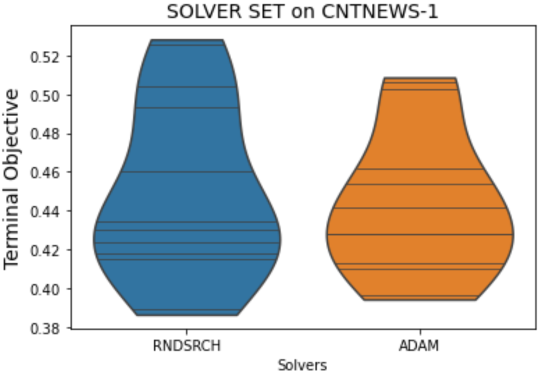
(a) for 10 macroreplications



(b) mean and CI

Figure: Unnormalized progress curves.

Terminal progress violin plots



What is SimOpt?

A testbed to enable finite-time comparisons of solvers, encourage solver development, and determine hard/do-able classes of problems

model A simulation model has a replicate method that generates one or more replications. Models can have many factors like c, s, w, α, β in continuous newsvendor

problem Built from a model, with objective function, decision variables, constraints

solver solvers tackle one or more problems, take simulation replications sequentially, and report a "running estimated best solution"

Problems

$$
\begin{aligned}
\min_x f(x, w) &= \mathbb{E} f(x, w, \xi) \\
\text{s/t } g(x, w) &= \mathbb{E} g(x, w, \xi) \leq 0 \\
h(x, w) &\leq 0 \\
x &\in \mathcal{D}(w).
\end{aligned}
$$

Problems have a problem-specific *budget* T measured in simulation replications

Solvers and the budget

- Solvers can be

  budget agnostic: They work the same irrespective of the budget T

  budget specific: They use the budget to set certain parameters, e.g., steplengths in SGD

- We mostly focus on budget-agnostic solvers today
- Some solvers use a random budget, e.g., R&S with statistical guarantees. Currently, we don't support random-budget solvers

Time, macroreplications, postreplications, bootstrapping

We measure time t through the fraction of the budget T that has been used, $t \in [0, 1]$

A macroreplication is a single run of a single solver on a single problem. The solver generates an estimated best solution as a function of time $(X(t) : 0 \leq t \leq 1)$

After the macroreplications are complete we use postreplications to estimate $(f(X(t)) : 0 \leq t \leq 1)$ for each macroreplication

Both macroreplications and postreplications are stochastic. We use bootstrapping to provide confidence intervals for $\mathbb{E}f(X(t))$ and related quantities at each $t \in [0, 1]$

Your turn

Generate unnormalized progress curves, aggregated unnormalized progress curves and terminal progress violin plots.

1. Run the GUI using the steps in the gdoc we shared
2. Click “Create Problem-Solver Group” and select the two solvers and “Max Profit for Continuous Newsvendor”
3. Click “Run” in the Workspace area below.
4. Click “Post Process and Normalize,” and in the new window use the defaults and click “Post-Process.”
5. Click “Plot”
6. From the new window, select the problem and two solvers, select the plot type, unselect the “Normalize by Relative Optimality Gap”, and click “Add” then “View Plot” (below)

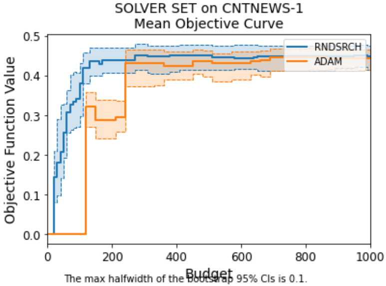
(Normalized) progress curves

- Recall that time is measured as a fraction of the budget T, so $t \in [0, 1]$
- Rescale the objective function to

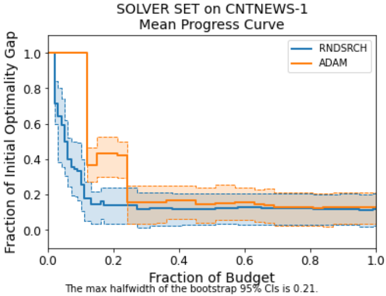
$$\frac{f(X(t)) - f(x^*)}{f(x_0) - f(x^*)}$$

- $f(x^*)$ = optimal objective function value, part of problem. If unknown, estimated as best solution seen

Normalized progress curves



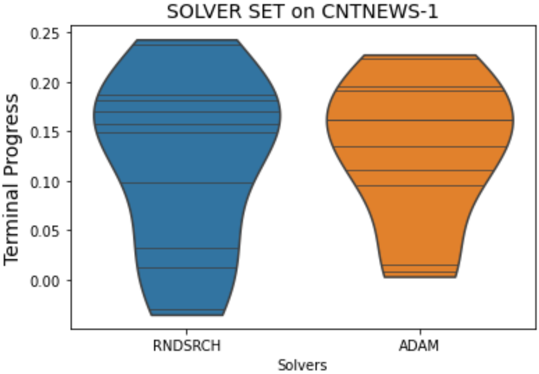
(a) unnormalized



(b) normalized

Figure: Mean+CI progress curves.

Terminal progress violin plots (normalized)

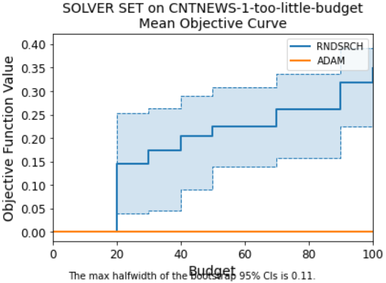


Your turn

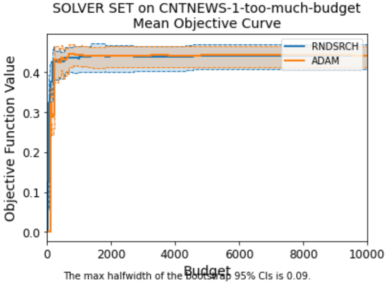
Generate normalized plots

1. From the Plot window, select the problem and two solvers, select the plot type and click “Add”
2. Click “View Plot,” which appears below

Bad budgets



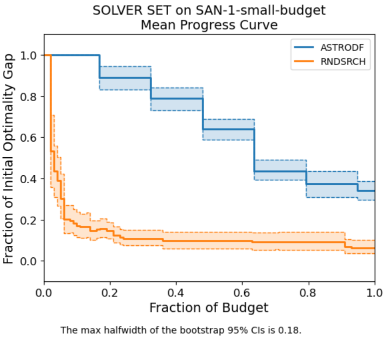
(a) Budget too small



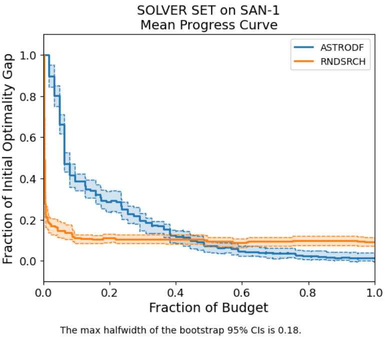
(b) Budget too large

Figure: Unnormalized aggregated progress curves.

Bad budgets (another example)



(a) Budget = 1000

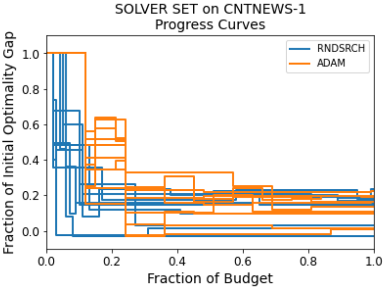


(b) Budget = 10,000

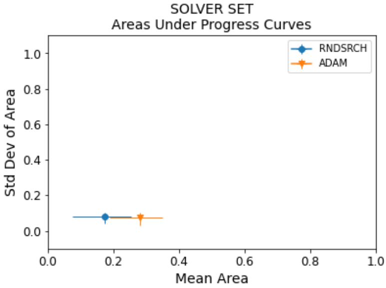
Figure: Normalized aggregated progress curves.

Area under progress curves and scatter plots

- Summarize performance of many solvers on many problems
- Let A be the (random) area under a normalized progress curve from a single macroreplication. Plot (μ_A, σ_A) for each problem and solver



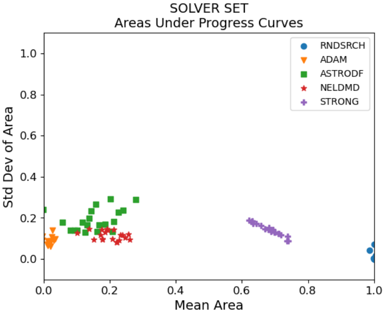
(a) aggregate progress curves



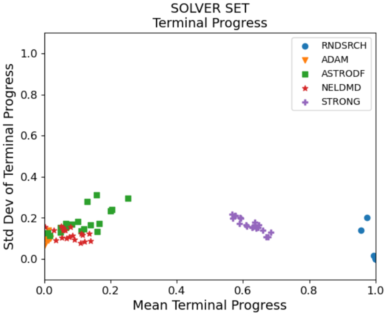
(b) scatter plots

Figure: Area under progress curves.

Scatter plots



(a) Area scatter plots



(b) Terminal scatter plots

Figure: Comparing 5 solvers on 20 problems.

Solvability profiles

Main purpose is to explore time-dependent performance of multiple solvers on multiple problems. First consider *one* problem and *one* solver

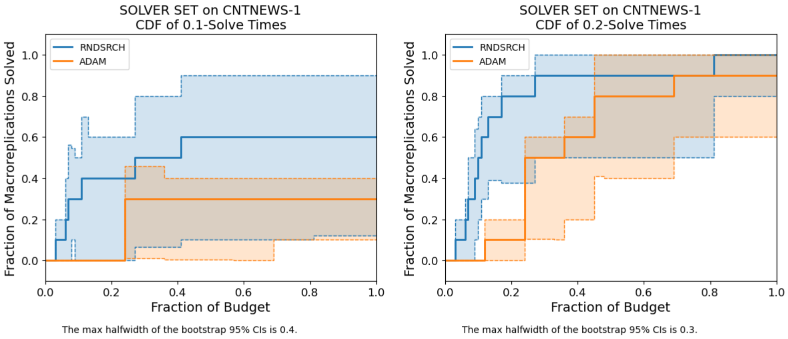
- Look at progress curves a different way. How long to reduce initial optimality gap to 10% of initial value?

$$\tau = \text{budget } t \text{ when first get within } 10\%$$

- τ is *random* and can $= \infty$ with positive probability
- We call τ the 10% solve time
- Could look at cdf and/or quantiles of τ ...

Solve time

One problem, Multiple Solvers



(a) $\alpha = 0.1$ (b) $\alpha = 0.2$

Figure: α-solve time CDF.

Solvability profiles

Multiple problems

Fix a solver

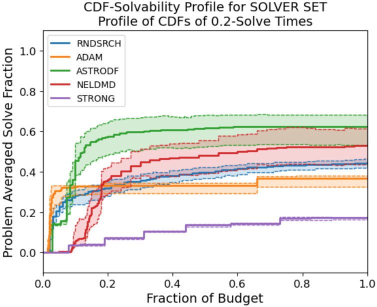
- Let τ^p be the solve time for problem p
- CDF solvability profile: As a function of t,

$$\rho(t) = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \Pr(\tau^p \leq t)$$

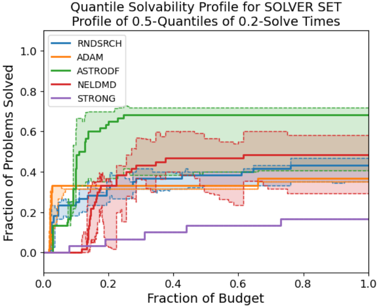
  Average, over problems, of probability solve by time t

- Quantile solvability profile gives the fraction of problems that are likely (a quantile) solved by time t, as function of t

Examples



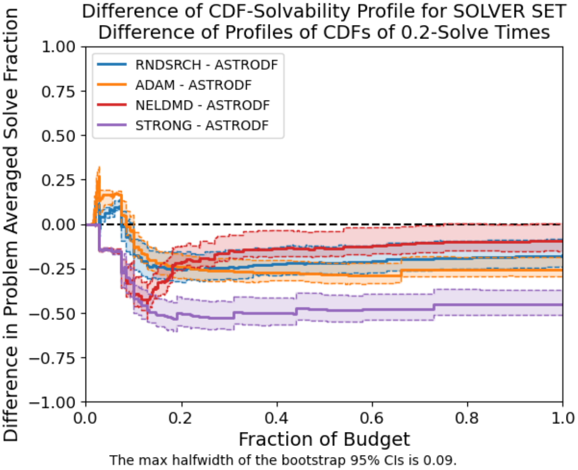
The max halfwidth of the bootstrap 95% CIs is 0.09.



The max halfwidth of the bootstrap 95% CIs is 0.18.

Difference profiles

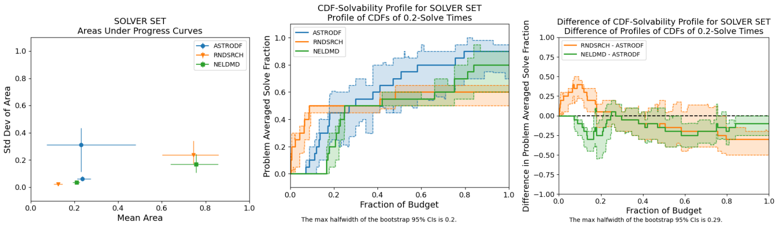
If you are focused on one solver in particular, compute differences in solvability profiles relative to that solver



Your turn

1. Create a new problem-solver group using problems IRONORECONT-1, SAN-1 and solvers ASTRODF, RNDSRCH, NELDMD
2. Run and postprocess the results. It'll take a few minutes; you can monitor progress in the Terminal panel.
3. Generate the plots you want.

Two problems, Three solvers



(a) Area scatter plots (b) Solvability profiles (c) Difference profiles

Figure: Comparing 3 solvers on 2 problems.

This ends Parts 1 and 2. On to Part 3!

Open the file `workshop/workshop.ipynb` in the VS Code editor and follow along.

Where to, next?

Thanks for attending today! Please stay involved! Here are some ideas:

1. Use SimOpt to test/improve your own solver, and share your solver with us
2. Use SimOpt to tackle your own problem, and share your problem with us
3. Tackle some of the items on the SimOpt to-do list (coming soon to GitHub)
4. Suggest improvements, bug fixes, ideas for new problems, ideas for new solvers

References

- http://simopt.org
- Diagnostic Tools for Evaluating and Comparing Simulation-Optimization Algorithms
  David J. Eckman, Shane G. Henderson and Sara Shashaani
  INFORMS Journal on Computing, 2022. To appear.
- SimOpt: A Testbed for Simulation-Optimization Experiments
  David J. Eckman, Shane G. Henderson and Sara Shashaani
  INFORMS Journal on Computing, 2022. To appear.