Redesigning a Testbed of Simulation-Optimization Problems and Solvers for Experimental Comparisons

David J. Eckman
Northwestern University (IEMS)

Shane G. Henderson
Cornell University (ORIE)

Raghu Pasupathy
Purdue University (Stats)

Winter Simulation Conference
December 9, 2019

Optimization Testbeds

In various optimization fields, **testbeds** have helped

- identify strong solvers;
- test and develop new solvers; and
- identify new research directions.

Examples: CUTEr/CUTEst, ALGLIB, MIPLIB, SIPLIB, Scikit-Optimize, COIN-OR, cORe.

Simulation-optimization (SO) community lags behind the deterministic-optimization community.

What is SimOpt?

SimOpt is a simulation-optimization testbed.

- Consists of a problem library and a solver library.
- All source code written in MATLAB.

Focus

Evaluating and comparing the **finite-time** performance of SO solvers.

Potential Uses:

- Develop a new solver and compare it to existing solvers.
- Devise rules for setting a solver's parameters.
- Determine which solvers work well on which classes of problems.

SimOpt Website

The website version (www.simopt.org) has several limitations:

- **Synchronization:** Hard to track and synchronize changes.
- **Submission:** Problem/solver submissions are handled offline.
- **Experimentation:** Wrapper function can run only one macroreplication of one solver on one problem. No plotting.

Functioned more as a library than as a testbed.

Guiding Principles

Maintainability: version control and synchronization.

- *Progress: transition the code library to GitHub.*

Usability: simple to download code and run experiments.

- *Progress: new wrapper functions.*

Controlled Experiments: standardized comparisons of solvers.

- *Progress: management of common random numbers.*

Versatility: can use the testbed for a variety of purposes.

- *Future work: testing different parameters, additional outputs.*

Variety: different problem and solver types.

- *Future work: integer-ordered variables, gradient-based methods, more submissions.*

GitHub

GitHub is a web-based platform for git version control.

Advantages:

- Linked wiki pages and README files.
- Ability to view code in browser.
- Tracking and reverting changes.
- Cloning/forking the repository.

New GitHub Repository

http://github.com/simopt-admin/simopt/wiki

Wiki Home Page

Welcome to SimOpt!

The purpose of the SimOpt testbed is to encourage development and constructive comparison of simulation-optimization (SO) solvers (algorithms). We are particularly interested in the finite-time performance of solvers, rather than the asymptotic results that one often finds in related literature.

For the purposes of this site, we define simulation as a very general technique for estimating statistical measures of complex systems. A system is modeled as if the probability distributions of the underlying random variables were known. Realizations of these random variables are then drawn randomly from these distributions. Each replication gives one observation of the system response, i.e., an evaluation of the objective function. By simulating a system in this fashion for multiple replications and aggregating the responses, one can compute statistics and use them for evaluation and design.

The paper Pasupathy and Henderson (2006) explains the original motivation for the testbed, and the follow-up paper Pasupathy and Henderson (2011) describes an earlier interface for MATLAB implementations of problems and solvers. The paper Dong et al. (2017) conducts an experimental comparison of several solvers in SimOpt and analyzes their relative performance. A forthcoming WSC paper (Eckman et al. 2019) describes in detail the recent changes to the architecture of SimOpt and the control of random number streams.

The Problem Library contains a variety of SO test problems and the Solver Library provides users with the latest SO solvers to solve different types of SO problems. You can also contribute new test problems and solvers to SimOpt by using pull requests in GitHub. The two libraries are intended to help researchers evaluate and compare the finite-time performance of existing solvers. Instructions on how to run solvers on problems can be found here.

▾ Pages 5

Find a Page...

- Home
- External Links
- Instructions
- Problem Library
- Solver Library

Clone this wiki locally

`https://github.com/simopt-a`

Problem Library

Problem Library

Abbreviation	Full Problem Name	Variable Class	Constraints
AMBUSQ	Ambulance Bases in a Square	Continuous	Variable Bounds
CNTNV	Continuous Newsvendor	Continuous	Variable Bounds
EOQ	Economic Order Quantity	Continuous	Variable Bounds
FACLOC	Facility Location	Continuous	Variable Bounds
GPE	Gamma Parameter Estimation	Continuous	Variable Bounds
MM1	Metamodeling of M/M/1 Call Center	Continuous	Unconstrained
QUEGG1	GI/G/1 Queue	Continuous	Variable Bounds
SAN	Stochastic Activity Network Duration	Continuous	Variable Bounds
TOLLNW	Toll Road Improvements in a Network	Continuous	Variable Bounds

Stochastic Activity Network Problem

README.mdown

(SAN) - Stochastic Activity Network Duration

About the Problem

Given a stochastic activity network, minimize the expected duration of the longest path from a starting node to an ending node.

For full details, see the documentation.

Properties

Variable Class: Continuous.

Constraints Class: Variable bounds.

Optimal Solutions: Unknown.

Known Structure: The objective function is convex. An IPA estimator of the gradient is provided in the code.

References

Avramidis, A. N., and J. R. Wilson (1996). Integrated variance reduction strategies for simulation. *Operations Research* 44(2):327-346. Paper.

Solver Library

Solver Library

Abbreviation	Full Solver Name	Variable Class	Constraints
ANDFER	Anderson-Ferris	Continuous	Unconstrained or Variable Bounds
GASSO	Gradient-Based Adaptive Stochastic Search for Simulation Optimization	Continuous	Unconstrained or Variable Bounds
KWCDLS	Kiefer-Wolfowitz SA with Central Differences and Line Search	Continuous	Unconstrained or Variable Bounds
NELDMD	Nelder-Mead	Continuous	Unconstrained or Variable Bounds
RANDSH	Random Search	Continuous	Unconstrained or Deterministic
SPSA	Simultaneous Perturbation Stochastic Approximation	Continuous	Unconstrained or Variable Bounds
STRONG	Stochastic Trust-Region Response-Surface Method	Continuous	Unconstrained or Variable Bounds

GASSO Solver

README.mdown

(GASSO) - Gradient-Based Adaptive Stochastic Search for Simulation Optimization

About the Solver

A gradient-based algorithm that iteratively draws candidate solutions from a sampling distribution and then, based on their objective function evaluations, updates the parameters of the sampling distribution. For more details, see the documentation.

Properties

Variable Class: Continuous.

Constraints Class: Unconstrained and variable bounds.

References

Zhou, E., and S. Bhatnagar. (2018). Gradient-based adaptive stochastic search for simulation optimization over continuous space. *INFORMS Journal on Computing* 30(1):154-167. Paper.

Submission Process

1. **User** creates new problem/solver files on their "forked" repository.
2. **User** commits and pushes changes to their remote copy.
3. **User** submits a "pull request" via GitHub web interface.
4. **Development team** makes line-item changes to the code until it works properly.
5. **Development team** merges changes with the master branch. New problem/solver files become accessible to all users.

Measuring Finite-Time Performance

Approach:

1. Fix a simulation budget (# of objective function evaluations).
2. Record the solutions that a solver would recommend, if it had to terminate at an intermediate budget.
   - E.g., the current (incumbent) solution.
   - E.g., the **estimated** best solution visited so far.
3. In a post-processing step, estimate the **true** objective function values at the recorded solutions.

The resulting curve *varies* from macroreplication to macroreplication.

Convergence Plot

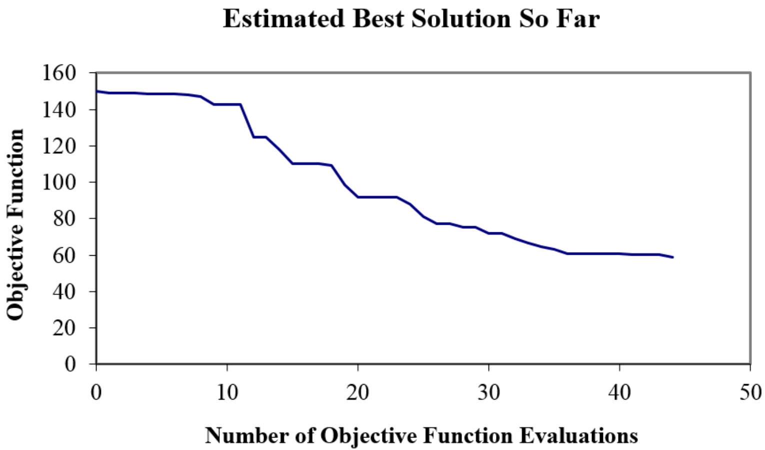

Figure: True objective function value of recommended solution over time.

Experimental Setup

Process for evaluating performance of one solver on one problem.

1. Run multiple macroreplications of the solver for a fixed budget.
2. Record recommended solutions at intermediate budget points.
3. Take fresh replications at recommended solutions to estimate objective function values.
4. At each budget point, calculate summary statistics (across macroreplications).
5. Make plots showing mean and median performance.

Wrapper Functions

RunWrapper.m *(Steps 1–2)* and PlotWrapper.m *(Steps 3–5)*.

Example: Stochastic Activity Network

Wrapper functions are called from the MATLAB terminal.

- Path must be set to the *Experiments* folder.

To run 30 macroreplications of several solvers on the SAN problem:

```
RunWrapper({'SAN'},{'ANDFER','GASSO','NELDMD'},30)
```

To run 100 post-replications at reported solutions and produce plots:

```
PlotWrapper({'SAN'},{'ANDFER','GASSO','NELDMD'},100)
```

Example: Stochastic Activity Network

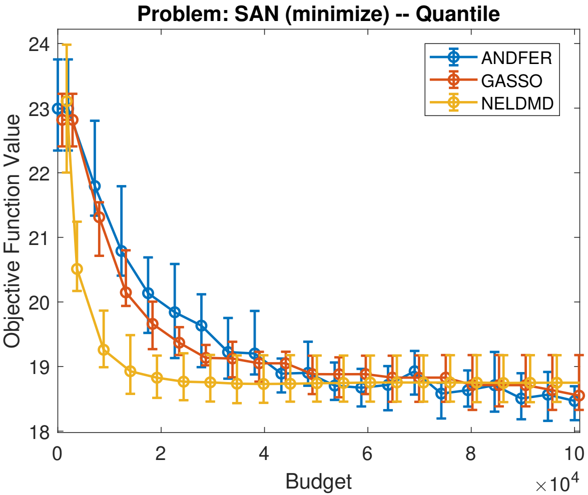

Figure: Median performance (+ quartiles) of select solvers on SAN problem.

Random Number Management

SimOpt functions uses random numbers for three purposes:

- **Solver:** Identifying solutions to evaluate.
- **Problem Structure:** Determining random initial solutions.
- **Problem:** Running simulation replications at a given solution.
  - Simulation model may use multiple sources of randomness.

SimOpt uses *streams* and *substreams* of `mrg32k3a` in MATLAB.

- Common random numbers (CRN) used for different purposes.

CRN in RunWrapper

Figure: Common random number schema for `RunWrapper.m` for two solvers (A and B). Sx.SSy denotes Substream y of Stream x.

		Internal	Initial	Solution 1	Solution 2
Solver A	Macrorep 1	(S1.SS1)	(S2.SS1)	Rep 1 (S3.SS1, S4.SS1)	Rep 1 (S3.SS1, S4.SS1)
				Rep 2 (S3.SS2, S4.SS2)	Rep 2 (S3.SS2, S4.SS2)
				Rep 3 (S3.SS3, S4.SS3)	Rep 3 (S3.SS3, S4.SS3)
				...	...
	Macrorep 2	(S5.SS1)	(S6.SS1)	Rep 1 (S7.SS1, S8.SS1)	Rep 1 (S7.SS1, S8.SS1)
				Rep 2 (S7.SS2, S8.SS2)	Rep 2 (S7.SS2, S8.SS2)
				Rep 3 (S7.SS3, S8.SS3)	Rep 3 (S7.SS3, S8.SS3)
				...	...
Solver B	Macrorep 1	(S1.SS1)	(S2.SS1)	Rep 1 (S3.SS1, S4.SS1)	Rep 1 (S3.SS1, S4.SS1)
				Rep 2 (S3.SS2, S4.SS2)	Rep 2 (S3.SS2, S4.SS2)
				Rep 3 (S3.SS3, S4.SS3)	Rep 3 (S3.SS3, S4.SS3)
				...	...
	Macrorep 2	(S5.SS1)	(S6.SS1)	Rep 1 (S7.SS1, S8.SS1)	Rep 1 (S7.SS1, S8.SS1)
				Rep 2 (S7.SS2, S8.SS2)	Rep 2 (S7.SS2, S8.SS2)
				Rep 3 (S7.SS3, S8.SS3)	Rep 3 (S7.SS3, S8.SS3)
				...	...

Future Work

- Grow SimOpt's problem and solver libraries.
- Changeable parameters (tuning and sensitivity analysis).
- Profiling solvers for comparisons *across* problems.

Acknowledgments

This material is based upon work supported by the Army Research Office under grant W911NF-17-1-0094 and by the National Science Foundation under grants DGE-165044, CMMI-1537394, and TRIPODS+X DMS-1839346.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.