07_so_with_simopt_lib

January 23, 2024

1 SO using the SimOpt library

1.1 M/M/1 Queue

Here we run a single replication of an M/M/1 queue and report several outputs of interest.

```
Model Parameters
----------------
The arrival rate is: 3.0
The service rate is: 8.0

Data collection
----------------
We allow a warm up of 20 people before calculating statistics based on 50
people.

Outputs
----------------
For a single replication:
Responses:
         avg_sojourn_time is 0.18.
         avg_waiting_time is 0.06.
         frac_cust_wait is 0.36.
```

1.2 Minimising the mean sojourn time in the M/M/1 queue

Here we introduce an objective function and solution space for a problem involving the M/M/1 queue.

The decision variable is the service rate μ.

There is a cost c associated with increasing the service rate μ.

```
 c: 0.1 is the cost per unit increase of service rate.
```

The objective $f(\mu; \xi, c) = y(\mu; \xi) + c^2$ where ξ is the random number stream, $y(\cdot)$ is the mean sojourn time. We wish to minimise the objective function $f(\cdot)$.

```
Solution space
----------------
The solution space (for the service rate mu) is continuous with lower bound 0
and upper bound inf
```

```
Model Parameters
----------------
lambda: 1.5 is the arrival rate.

Data collection
----------------
We allow a warm up of 20 people before calculating statistics based on 50
people.
```

We now run 10 simulation replications for a given solution and report the objective function values.

```
Ran 10 replications of the MM1-1 problem at solution x = (5,).

The mean objective estimate was 2.793 with standard error 0.0124.
The individual observations of the objective were:
         2.7808
         2.8516
         2.7651
         2.7641
         2.7547
         2.7358
         2.8083
         2.8355
         2.837
         2.7968
```

1.2.1 How the solvers work

Each solver is run a number of times (set as the number of macro-replications) on the problem. Each macro-replication involves a different random number stream. During each macroreplication, a solver will start with an initial solution (the initial 'current best') and at certain points in the simulation budget will update its recording of the current best solution. The 'progress curve' for a given macro-replication is the set of current best solutions along with the point in the simulation budget where those solutions became the current best. Post-replications are performed on each solution in each progress curve.

The solver then selects the optimal solution as the best of all solutions which were post-replicated. Further post-replications are performed on the initial and optimal solution - this is used to calculate an optimality gap which is used for normalising the progress curves of the solver.

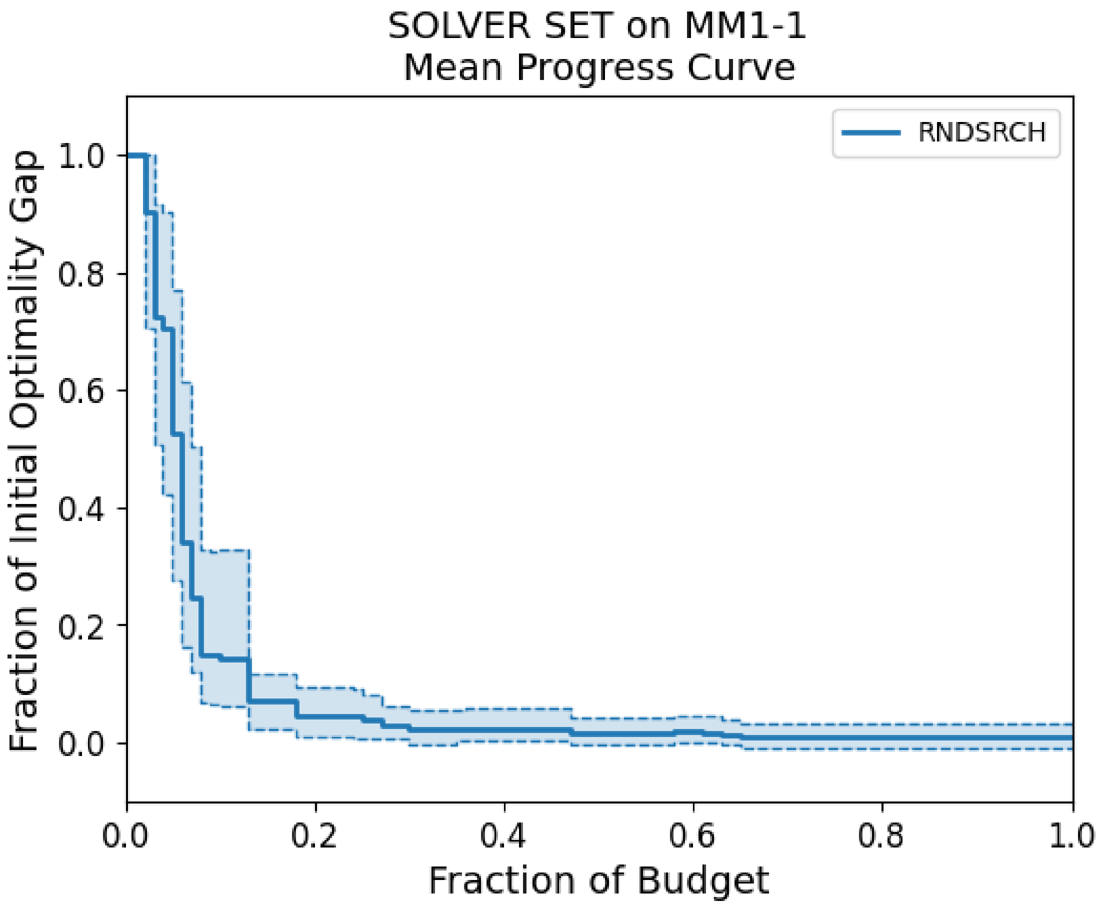
Bootstrap samples of the current best solution at different points of the budget and the objective values of those solutions. CI intervals are applied to plot the progress of the algorithm over the course of the simulation budget.

1.2.2 The Random Search solver

The solver starts at a given initial solution (which starts of as the 'current best') and then at each iteration moves to a randomly selected new solution. New random solutions are random samples from exponential distribution with rate 1/3. At each selected solution, 10 simulation replications are performed, and the mean objective value is calculated. Whenever a solution's mean objective is better than the current best, it is updated to be the current best. This process is repeated, using common random numbers at each solution, until the simulated budget is exhausted.

Below is a summary of the model factors, problem factors and solver factors, and high level details of the results of the experiment using the random search solver on the problem with the M/M/1 queue. The subsequent plot illustrates the performance of the solver as simulation budget is expended. The performance is quantified in terms of the fraction of the optimality gap remaining versus how much simulation budget has been used. The confidence intervals are calculated using bootstrapped samples of the current best solution at different points of the budget and the objective values of those solutions.

```
Solver: RNDSRCH

Model Factors:
        warmup: 50
        people: 200
        lambda: 1.5


Problem Factors:
        initial_solution: (5,)
        budget: 1000
        cost: 0.1


Solver Factors:
        crn_across_solns: True
        sample_size: 10


10 macroreplications were run.

100 postreplications were run at each recommended solution.


The initial solution is (5,). Its estimated objective is 2.7844.

The proxy optimal solution is (2.8243,). Its estimated objective is 1.5488.

100 postreplications were taken at x0 and x_star.



 ['experiments/plots/SOLVER_SET_on_MM1-1_mean_prog_curve.png']
```



We next use two other solvers (alongside the random search solver) to solve the problem based on the M/M/1 queue. The solvers we have chosen are called 'STRONG' and 'ADAM'. Below we give a brief description of how each of these solvers works, and the subsequent plot illustrates the performance of the solver as simulation budget is expended. The performance is quantified in terms of the probability that the solver can come close to the optimal solution given a certain proportion of budget expenditure. 'Close to optimal' is defined as within 10% (of the initial optimality gap) away from the optimal solution. The confidence intervals are calculated using bootstrapped samples of the current best solution at different points of the budget and the objective values of those solutions.

1.2.3 STRONG solver

STRONG (a stochastic trust-region response surface method) is a meta-modelling approach to simulation optimisation where local meta-models (which are fitted to simulation data) include first-order and quadratic models, depending on the size of the 'trust-region'. The trust region can be considerred the local area within the solution space over which the meta-models are optimised at each iteration.

1.2.4 ADAM solver

ADAM is a stochastic gradient descent method which works with estimates of the first and second moments of the gradient of the objective function. At each timestep these estimates are updated as moving averages using new estimates of the gradient at the incumbent solution from IPA or Finite-Difference estimators. A new solution is found by moving along the gradient by an amount dictated by the step-size.

```
['experiments/plots/SOLVER_SET_on_PROBLEM_SET_profile_cdf_0.1_solve_times.png']
```