A practical approach to subset selection with chance constraints

Thomas Monks [1]   Christine S.M Currie [2]

[1]NIHR CLAHRC Wessex, UoS / Alan Turing Institute

[2]CORMSIS, University of Southampton

Winter Simulation Conference 2018

Overview

Motivation

Aim: support system experts making complex decisions involving **multiple objectives** and a **large number of scenarios** for the system

Factors:

- Some unquantifiable (e.g., political) variables
- Large, complex, slow-running simulation model
- Simulation practitioner without a PhD in statistics/simulation
- Off-the-shelf simulation package

Motivation

Aim: support system experts making complex decisions involving **multiple objectives** and a **large number of scenarios** for the system

Factors:

- Some unquantifiable (e.g., political) variables: Find a subset not a single optimum
- Large, complex, slow-running simulation model: Use variance reduction techniques, e.g., CRN
- Simulation practitioner without a PhD in statistics/simulation: Reduce the need for expert statistical judgment
- Off-the-shelf simulation package: Difficult to implement fully sequential methods

Motivation

Requirements

R1 The choice of options to include in the experimentation and the number of replications to make can only be changed once during the experiment (two-stage method)

R2 The procedure should not impose any distributional assumptions on the simulation output

Problem Description

Assume that we are comparing $k \geq 2$ systems and are primarily interested in minimizing the mean value of a particular output

$$x_i = \sum_{j=1}^{n} x_{ij}/n$$

where $i = 1, \ldots, k$ but are also interested in L secondary outputs or objectives

$$y_{il} = \sum_{j=1}^{n} Y_{ijl}/n$$

Subset Selection with Chance Constraints

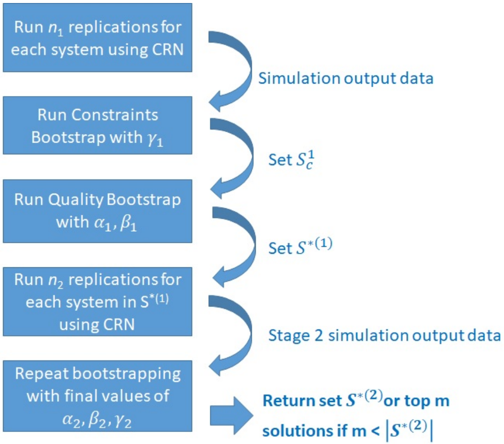
Aim: Identify a shortlist (subset) of systems ($\mathbf{S}^*$) that are all within a **proportion** β of the best system with probability $1-\alpha$
And satisfy the chance constraints with a probability $1-\gamma$

We restrict the number of systems on the shortlist to $\min\{m, |S^*|\}$ by taking the **top m** systems that satisfy the above constraints

Previous Work

1. **Approach:** [Branke et al. 2007] suggest three categories: indifference zone, OCBA and Expected Value of Information
2. **Chance constraints** [Hong et al. 2015] suggest two approaches to dealing with chance constraints: Expectation Constrained Selection and Chance Constrained Selection.
3. **Subset selection** authors use either OCBA or indifference zone methods to maximize/guarantee the probability of correct selection of the best m of k systems

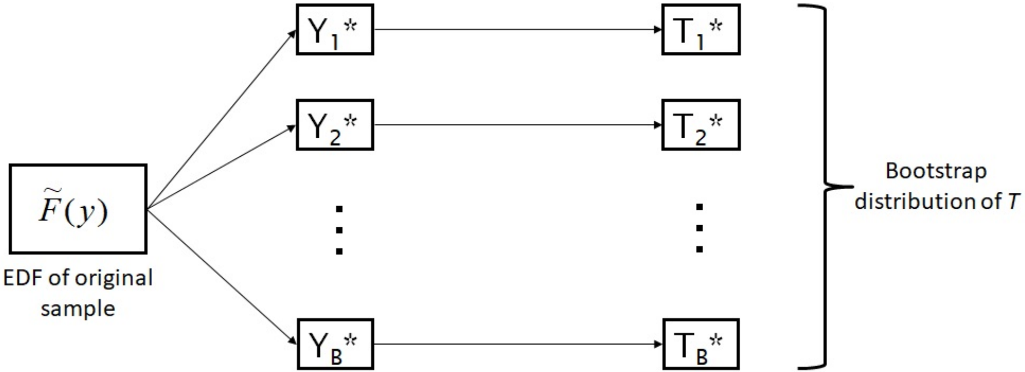
Big Picture



- Method relies on bootstrapping
- Set stage 1 parameters so that we are risk averse
- Balancing risk of missing a good solution versus including too many in stage 2
- Trade off between n_1 and n_2

Non-Parametric Bootstrapping

Resampling method used to infer properties for a set of data.



Constraints Bootstrap

Aim: Identify systems likely to violate the chance constraints

Constraints Bootstrap

1. Input a set of bootstrap samples $\mathbf{Y}^{\star(1)}, \mathbf{Y}^{\star(2)}, \ldots, \mathbf{Y}^{\star(B)}$ and for each calculate $y_l^{\star(b)}$, $l = 1, \ldots, L$.
2. Include systems in the final feasible set $\mathbf{S}_c$ if

$$\frac{1}{B} \sum_{b=1}^{B} \prod_{l=1}^{L} I \left\{ y_l^{\star(b)} \geq 0 \right\} \geq 1 - \gamma,$$

3. Return $\mathbf{S}_c$.

Quality Bootstrap

Aim: identify a set of systems with means within a distance β of the best system with probability $1 - \alpha$

Quality Bootstrap

1. Define a new variable $d_{ij} = x_j^* - x_{ij}$
2. Generate B bootstraps of the d_{ij}
3. In each bootstrap sample, identify systems with differences less than $\beta \bar{x}^*$, where $\bar{x}^*$

Quality Bootstrap

Quality Bootstrap (Cont'd)

4. Identify $\mathbf{S}^*$ such that it is the biggest set for which

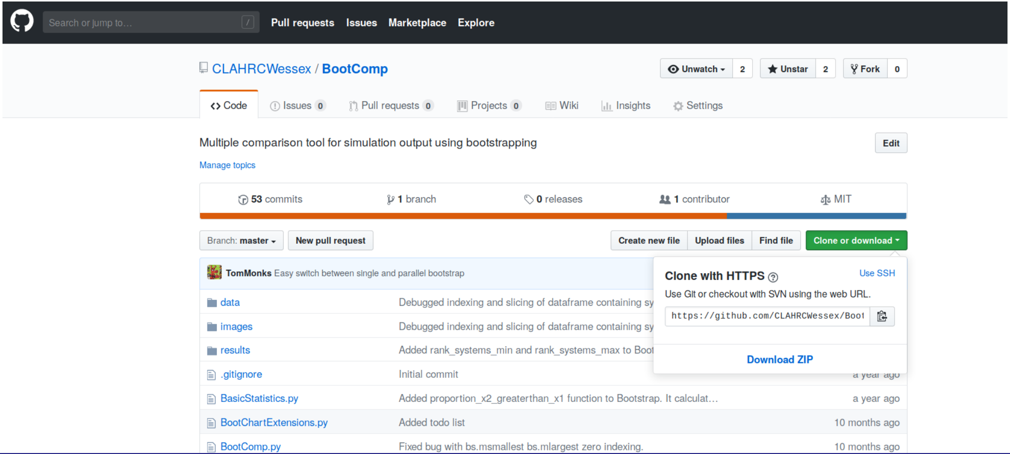
$$\frac{1}{B}\sum_{b=1}^{B}\prod_{j\in\mathbf{S}_c} I\{|d_{ij}^{\star(b)}| \leq \beta\bar{x}^*\} \geq 1-\alpha.$$

5. Return $\mathbf{S}^*$

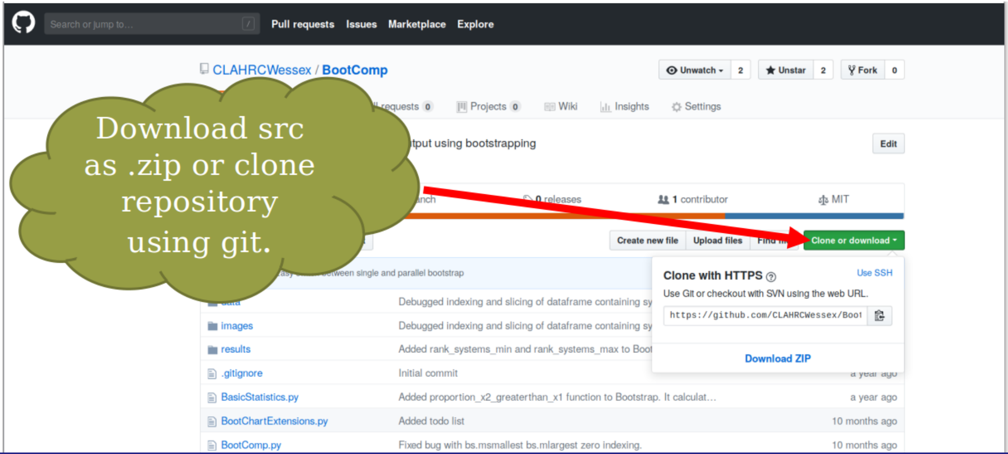
Advice on installation of Python

https://www.anaconda.com/download/

Code available from GitHub



Code available from GitHub



Jupyter Notebook Implementation

jupyter BootComp_WSC18 Last Checkpoint: Yesterday at 11:34 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Markdown

1.2. Output data

The output data for the example analysis are bundled with git repository. There are three .csv files in the data/ directory for 'waiting times', 'utilization' and 'transfers'.

The model itself is not needed. There are 50 replications of 1151 competing designs points. Users can vary the number of replications used in the two stage procedure.

The experimental design is also included for reference.

2. Prerequisites

2.1. BootComp Modules for WSC18 Paper

In [1]:

```
import helper_wsc18 as wsc
from Bootstrap_wsc18 import constraints_bootstrap, quality_bootstrap
from helper_wsc18 import best_subset_table, get_best_subset, load_model_file
```

Jupyter Notebook Implementation

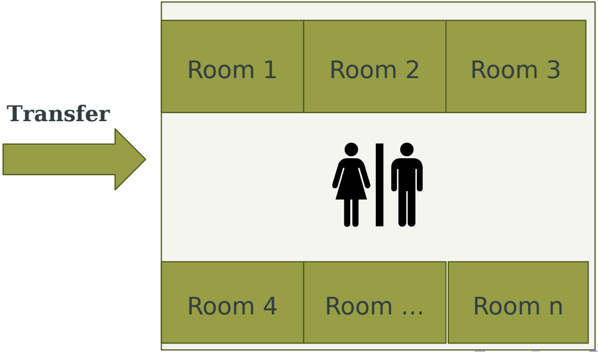
Dependencies

- The readme.md provides an install guide (read the readme!)
- Create a conda environment
- Environments allow you to switch versions of Python packages to make sure you are using the same dependencies as the original code

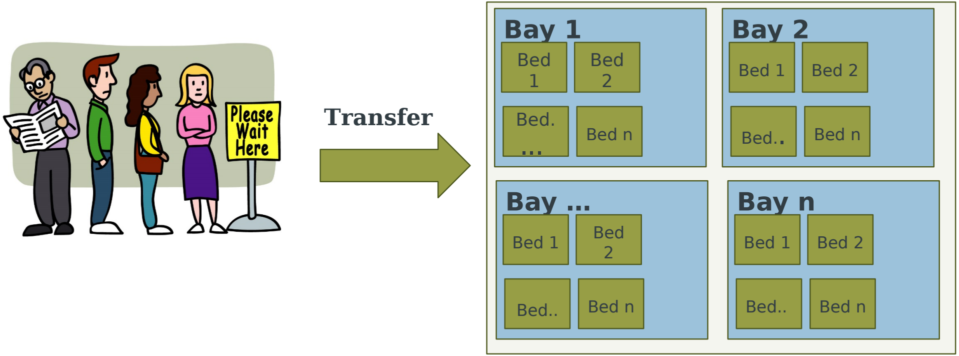
```
conda env create -f environment.yml
conda activate bootcomp
```

Designing a rehabilitation ward

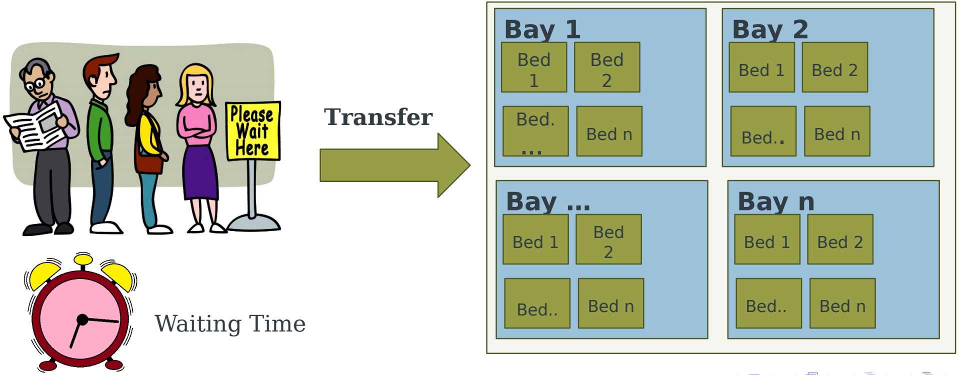
Patients waiting in an acute hospital for transfer to rehabilitation



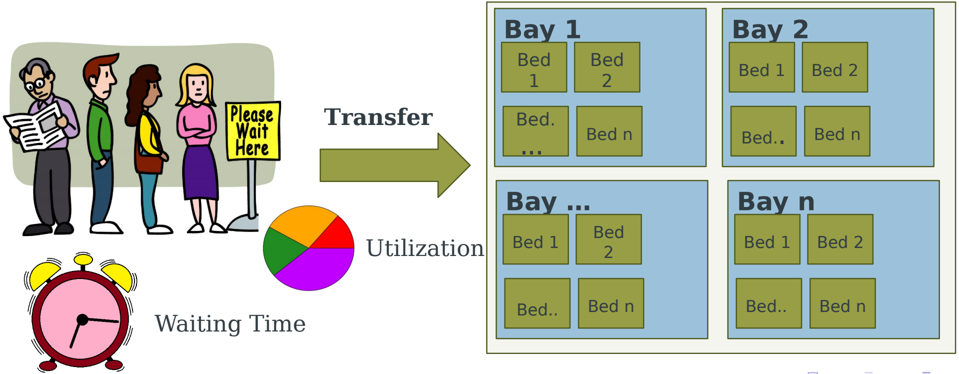
Designing a rehabilitation ward



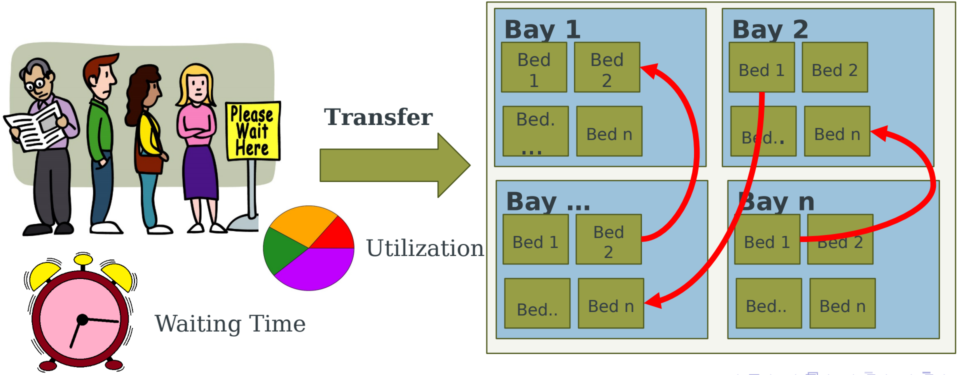
Designing a rehabilitation ward



Designing a rehabilitation ward



Designing a rehabilitation ward



Design of Experiments

1051 competing designs

Decision Variables

- No. beds
- Bay size + no. bays
- No. single rooms

Chance Constraints

- Utilization of beds
- Patient transfers between bays

Further work

- Comparison with sequential budget allocation algorithms finding top-m systems with a **single** performance measure
  - Set of 10 normal distributions $N(i, 6)$
  - Set of 100 normal distributions $Ni/100, 6)$
  - Law inventory example
- Comparisons using common random numbers
- Identifying “good” values for the parameters in the first stage: $N_0, \alpha_1, \beta_1, \gamma_1$

References

Jürgen Branke, Stephen E. Chick and Christian Schmidt (2007)
Selecting a selection procedure
Management Science 53, 1916–1932

L. Jeff Hong, Jun Luo and Barry L. Nelson (2015)
Chance constrained selection of the best.
INFORMS Journal on Computing 27, 317 – 334.