

## **PRACTICAL CONSIDERATIONS IN SELECTING THE BEST SET OF SIMULATED SYSTEMS**

Thomas Monks

NIHR CLAHRC Wessex  
Southampton General Hospital  
Southampton, SO16 6YD  
UK

Christine S.M. Currie

Mathematical Sciences  
University of Southampton  
Highfield  
Southampton, SO17 1BJ  
UK

### **ABSTRACT**

In many practical problems, simulation models are used to support complex decision-making processes comparing hundreds or thousands of solutions. These problems typically have a key objective but the final decision may be dependent on other factors, which cannot be incorporated into the simulation model. In such cases, decision-makers may request a short list of ‘good’ solutions, which work well for the main objective and satisfy one or more chance constraints. While fully sequential ranking and selection procedures can be effective at solving these problems, surveys of experimentation practice suggest that they are under-utilized, potentially due to difficulties automating commercial software. We develop an approach with just two stages of replications. The approach, which has been designed to cope with the use of common random numbers, draws on ideas from indifference zones and makes use of bootstrapping to find a subset of high quality solutions. A Python implementation is freely available.

### **1 INTRODUCTION**

Simulation models are often used to whittle down a large number of potential solutions to a short list of high quality solutions that satisfy any constraints imposed by the decision-maker and perform well on the main objective. This allows the decision-maker to work with these “good” solutions to identify which will work best in the real-life system, as often other factors come into play in making the final decision, which it is not possible to include in the simulation model. Our aim in this research is to solve this problem but in a way that makes it accessible to a simulation modeler with a working knowledge of statistics and a limited simulation package. This places two requirements on the method.

- 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, effectively making this a two-stage process.
- R2 The procedure should not impose any distributional assumptions on the output.

Requirement R1 is included based on practical experience, which suggests that, although a few simulation software packages include fully sequential ranking and selection procedures, the majority do not. As a result, most simulation practitioners are left with the rather daunting task of incorporating these themselves. Not only is this complex, but in the instances where simulation software must be externally automated there may also be an efficiency penalty. For example, in time needed for communication between ranking and selection code and the simulation software, updating the model parameters and (re)initializing the model. Although we insist on a fairly restricted set up, the bootstrap ranking and selection procedures developed by Lee and Nelson may be particularly beneficial here as they suggest conducting  $\Delta n$  observations at each step rather than just one (Lee and Nelson 2014; Lee and Nelson 2015). Balancing the loss of accuracy of using a high value of  $\Delta n$  with the computation cost of communications between the optimization software

and the simulation software could be an interesting practical problem but goes beyond the scope of this article.

At first sight, R1 seems to imply that we ignore much of the recent research and return to the literature from around 15 years or more ago, such as Chick and Inoue (2001) and earlier articles such as Koenig and Law (1985), Gupta (1965), Nelson and Matejcik (1995). In actual fact, there is a great deal in the more recent ranking and selection literature that can benefit us with the two-stage problem as we discuss in the literature review.

Requirement R2 is important from a practical perspective as it ensures that the methods we develop are robust to the character of the output. We also assume that common random numbers (CRN) will be used in the simulation experiments, as this is standard practice in all main stream simulation packages, but the method does not rely on them being used. We use bootstrapping for our analysis, a relatively straightforward way of dealing with non-standard data. Cheng and Currie (2009) give an introduction to the use of bootstrapping methods in the analysis of simulation models but for a more comprehensive description of analyzing non-standard data, we recommend Cheng (2017).

Optimizing a single variable with chance constraints can be a way of simplifying a multi-criteria decision problem, as discussed in Hong et al. (2015), with the secondary objectives expressed as chance constraints. This is how we set up the practical problem that we consider in Section 5, preferring this approach to the alternative of finding a Pareto front of solutions (Lee et al. 2010). This partly reflects the nature of the problem we are solving but is also due to the ease of explaining the ideas to a decision-maker.

The literature review in the next section describes how our procedure draws on previous work in the ranking and selection literature. We make use of the nonparametric bootstrap to carry out statistical analysis of the output and the basis for this procedure is also described here. In Section 3 we set out the problem in a more precise way before going on to describe our proposed procedure in Section 4. The work was motivated by a real-life simulation project, set up to support the design of a new community hospital rehabilitation ward and we describe the project and results in Section 5.

## 2 LITERATURE REVIEW

Traditional ranking and selection problems aim to find one of a finite set of systems that maximizes a single output measure, where the output measure is a stochastic variable. Early algorithms considered two-stage procedures, as we do here, while more recent work has developed fully sequential methods. (Branke et al. 2007) split ranking and selection algorithms into three main approaches:

- (i) **Indifference Zone** procedures such as KN++ (Kim and Nelson 2006a), which guarantee that the probability of correct selection will be greater than a user-defined threshold, assuming that the mean output of the best system is sufficiently distant from the output of the next best system (outside the indifference zone).
- (ii) **Expected Value of Information** procedures rely on Bayesian statistics and allocate samples so as to maximize the expected value of information coming from them.
- (iii) **Optimal Computing Budget Allocation** procedures were initially introduced by Chen (1996) and sequentially allocate samples so as to improve the estimate of the probability of correct selection, again using Bayesian statistics.

We refer the interested reader to Kim and Nelson (2006b), Branke et al. (2007), Fu (2015), all of which include excellent reviews and analyses of ranking and selection algorithms.

### 2.1 Chance Constraints

Andradóttir and Kim (2010) provide the first formulation of a constrained ranking and selection problem, in which they introduce an indifference zone approach to identifying the best system that satisfies a single constraint. While they assume that the main and secondary outputs are normal, the results presented seem

relatively robust to non-normality. Their approach to the chance constraint relies on Bechhofer’s indifference zone structure (Bechhofer 1954) in which three different regions are identified for the value of the chance constraint: desirable, acceptable and unacceptable. Similar to our work, they use the expected value of the secondary output measure in the constraint equation. Hong et al. (2015) describe this as Expectation Constrained Selection of the Best (ECSB) and introduce a method that instead solves the problem of Chance Constrained Selection of the Best (CCSB). In CCSB, the constraints on the secondary outputs take the form of probabilistic constraints, meaning that the constraints imposed by the decision-maker must be satisfied with a given probability. Imposing this structure on the constraints facilitates checking the feasibility of systems. The approach that Hong et al. (2015) introduces also assumes normality for the simulation outputs and selects just one optimal value. It allows for CRN and finds a relatively conservative solution for the number of replications due to its assumption that all chance constraints must be satisfied subject to the Bechhofer indifference zone structure.

Chance constraints have also been included in algorithms that aim to optimally allocate a fixed simulation budget in ranking and selection problems. For example, Pasupathy et al. (2014) make an elegant use of bilevel optimization to determine an algorithm SCORE to solve the ECSB problem; and Hunter and Pasupathy (2013) put forward a sequential algorithm that maximizes the asymptotic decay of the probability of incorrect selection, reliant on knowing the underlying distribution of the output data.

## 2.2 Subset Selection

Defining a good shortlist of solutions can be subjective. Much of the literature on optimal subset selection aims to identify the best  $m$  options out of a total of  $k$  possibilities, where the quality of an option is measured by its average output over all of the simulation replications. A measure of how well the method can identify these top solutions is the probability of correct selection (PCS), and methods will either aim to maximize it within a fixed computational budget (e.g., Gao and Chen (2016), Chingcuanco and Osorio (2013)) or find the number of replications required to guarantee that it exceeds a minimum value (e.g., Koenig and Law (1985), Wang et al. (2011)). Gao and Chen (2015) instead minimize the Expected Opportunity Cost (EOC), measured as the difference between the selected designs and the best designs. The use of EOC as a measure of the quality of a ranking and selection procedure was first introduced by Chick and Inoue (2001) and has a clear practical benefit.

None of the current methods consider the rank order within the subset, instead focusing on identifying the correct subset of options.

## 2.3 Bootstrap Resampling for Ranking and Selection

Recent work has used bootstrapping to enable ranking and selection procedures to work with a wider range of distributions (Bekki et al. 2010; Lee and Nelson 2016) and to improve the computational efficiency of the comparisons (Lee and Nelson 2015). We draw on these ideas in the bootstrap procedures developed in Section 4.

## 3 PROBLEM DESCRIPTION

Assume that there are in general  $k \geq 2$  systems being compared and define  $X_{ij}$  to be univariate, real-valued output data coming from replication  $j$  for system  $i$  that represents the main performance measure. We make no assumptions about the distribution of the  $X_{ij}$  and allow for correlations between observations from different systems in the same replication, which may arise if CRN are being used. Our main performance measure is assumed to be the mean of the output for a particular system after  $n$  replications,

$$x_i = \mathbb{E}(X_{ij}) = \sum_{j=1}^n X_{ij}/n,$$

and we assume here that small values of  $x_i$  are desirable.

We also define a set of  $l = 1, \dots, L$  secondary outputs,  $Y_{ijl}$  coming from replication  $j$  for system  $i$ . The  $Y_{ijl}$  are real-valued and we assume that their mean values,

$$y_{il} = \mathbb{E}(\mathbf{Y}_{ijl}) = \sum_{i=1}^n \mathbf{Y}_{ijl} / n.$$

Our problem is one of subset selection with chance constraints on the secondary outputs. We wish to identify a shortlist or subset of systems  $\mathbf{S}^*$  that are all within a proportion  $\beta$  of the mean of the best system,  $x^*$  with a probability  $1 - \alpha$  and fail the chance constraints with probability less than  $\gamma$ . This is an indifference zone approach but we define the indifference zone as a proportion of the best solution (e.g., within 10%) rather than giving it an absolute value. A system is said to satisfy the chance constraints if  $\mathbb{P}\{\mathbf{y}_i \geq \mathbf{0}\} \geq 1 - \gamma$ , where we use  $\mathbf{y}_i$  to represent the vector of expected values  $y_{il}, l = 1, \dots, L$ . Combining, our problem can be written as one of determining  $\mathbf{S}^*$  such that

$$\begin{aligned} \mathbb{P}\left\{\bigcap_{i \in \mathbf{S}^*} (x_i \leq x^*(1 + \beta))\right\} &\geq 1 - \alpha, \\ \mathbb{P}\{\mathbf{y}_i \geq \mathbf{0}\} &\geq 1 - \gamma, \quad i \in \mathbf{S}^* \end{aligned} \quad (1)$$

If the size of  $\mathbf{S}^*$ ,  $|\mathbf{S}^*| > m$ , the maximum length of the shortlist, the top  $m$  systems in a ranked list of mean outputs of the key measure will be included in the shortlist. We make no claims that these will be the top  $m$  methods, but only that they all lie within  $\beta$  of the best and have a high probability of being feasible. From a statistical perspective, this is the least satisfactory part of the method, but it helps in situations where there are a large number of suitable solutions. An alternative would be to reanalyze the results with smaller values for  $\beta$  or  $\alpha$ .

## 4 METHODS

The method relies on two bootstrap routines, which are both run twice: first in stage one and again at the end of stage two. The first, **Constraints Bootstrap** is used to determine the probability of a particular system violating the constraints, while the second, **Quality Bootstrap** identifies the systems with means within a proportion  $\beta$  of the mean of the best remaining systems. We begin by describing the whole procedure before giving more detailed descriptions of the two bootstrap routines.

1. Run  $n_1$  replications of the simulation model for all systems using CRN if available to generate a set of primary outputs  $X_{ij}$  and secondary outputs  $Y_{ijl}$ , where  $j = 1, \dots, n_1; i = 1, \dots, k; l = 1, \dots, L$ .
2. Remove any systems that have a probability of failing the chance constraints of greater than  $\gamma_1$  leaving a set of systems  $\mathbf{S}_c^1$  that satisfy the chance constraints after the first stage of sampling. The set  $\mathbf{S}_c^1$  is identified via bootstrapping using the **Constraints Bootstrap** described below in Section 4.1 with inputs  $\mathbf{S}$ , the complete set of systems,  $\mathbf{Y}$ ,  $\gamma_1$ . Label the output  $\mathbf{S}_c^1$ .
3. Run the **Quality Bootstrap** described below in Section 4.2 with inputs  $\alpha_1, \beta_1, \mathbf{S}_c^1$  and the  $n_1$  by  $k$  matrix of results for the main output measure  $\mathbf{X}$ . This outputs the set of systems  $\mathbf{S}^{*(1)}$  that have means within a proportion  $\beta_1$  of the best mean with probability greater than  $1 - \alpha_1$ .
4. Run  $n_2$  replications of the simulation model for all systems in the set  $\mathbf{S}^{*(1)}$  using CRN if available.
5. Follow steps 2 and 3 using data from both stages in  $\mathbf{X}$  and  $\mathbf{Y}$  and parameters  $\gamma_2, \alpha_2, \beta_2$ . This results in a final shortlist of feasible and high quality systems  $\mathbf{S}^{*(2)}$ .
6. If the number of systems is bigger than the desired number  $m$ , pick the top  $m$  in a non-decreasing list of means of the main output measure. Ties are decided using secondary output measures  $\mathbf{Y}$ .

We use different values for  $\gamma_1$  and  $\gamma_2$  in the numerical results but keep  $\alpha$  and  $\beta$  the same in both stages. Determining the appropriate values for these parameters in the first stage needs further investigation.

#### 4.1 Constraints Bootstrap

Our aim in the Constraints Bootstrap is to identify systems that are likely to violate the chance constraints. No comparisons are made between systems but we assume that we have more than one chance constraint. We use a fairly conservative measure of satisfying the chance constraints, as suggested in Hong et al. (2015) in which we insist that *all* of the constraints are satisfied simultaneously.

The routine takes as inputs  $\mathbf{Y}$ , the secondary outputs of the simulation model such that  $Y_{ijl}$  is the value of the  $l$ th chance constraint,  $l = 1, \dots, L$  in the  $j$ th replication for the  $i$ th system. Here,  $i \in \mathbf{S}$  and  $j = 1, \dots, n$ . Also input is  $\gamma$ , the threshold probability for failing the chance constraints.

As is standard with bootstrapping, we assume that  $\mathbf{Y}$  can be viewed as a sample from a true multivariate distribution  $G$  and use bootstrapping to sample from an approximation to this distribution,  $\hat{G}$ . The bootstrapping proceeds as follows.

1. Generate  $B$  bootstrap samples of size  $n$  using a non-parametric bootstrap in which we sample with replacement from  $\mathbf{Y}$ . This yields a set of bootstrap samples  $\mathbf{Y}^{*(1)}, \mathbf{Y}^{*(2)}, \dots, \mathbf{Y}^{*(B)}$  and for each of these we calculate  $y_l^{*(b)}$ ,  $l = 1, \dots, L$ .
2. Include systems in the final feasible set  $\mathbf{S}_c$  if

$$\frac{1}{B} \sum_{b=1}^B \prod_{l=1}^L I\{y_l^{*(b)} \geq 0\} \geq 1 - \gamma, \quad (2)$$

where  $I(\cdot)$  is the indicator function, which takes a value of 1 if the condition inside the parentheses is true and 0 otherwise.

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

#### 4.2 Quality Bootstrap

The quality bootstrap aims to identify a set of systems  $I^*$  that all have means within a distance  $\beta$  of the mean of the best sampled system with a probability  $1 - \alpha$ . It takes as inputs  $\alpha, \beta$ , the set of feasible systems  $\mathbf{S}_c$ , and  $\mathbf{X}$ , a set of outputs for the main output measure  $X_{ij}$ ,  $i \in \mathbf{S}_c$ ,  $j = 1, \dots, n$ , where  $n$  is the number of replications being considered in the bootstrap.

1. Define a new variable  $\mathbf{d}$  with elements  $d_{ij}, i \in \mathbf{S}_c, j = 1, \dots, n$ , such that

$$d_{ij} = x_j^* - X_{ij},$$

where  $x_j^*$  is the value of the main output measure for the best, feasible system in stage 1 in replication  $j$  of the simulation. We can treat the set of  $d_{ij}$  as a sample from the true multivariate distribution for the differences,  $F$ .

2. Generate  $B$  bootstrap samples of size  $n$  from  $\hat{F}$ , our approximation to  $F$ . We do this using a nonparametric bootstrap in which we sample with replacement from  $d_{ij}$ . This gives us a new set of samples  $\mathbf{d}^{*(1)}, \mathbf{d}^{*(2)}, \dots, \mathbf{d}^{*(B)}$ , where each  $\mathbf{d}^{*(b)}$  is of the same dimensions as the original set of differences,  $\mathbf{d}$ .
3. In each bootstrap sample, identify systems with differences less than  $\beta \bar{x}^*$ , where  $\bar{x}^*$  is the mean of the key output measure for the best system in the original data.
4. Identify  $\mathbf{S}^*$  such that it is the biggest set for which

$$\frac{1}{B} \sum_{b=1}^B \prod_{j \in \mathbf{S}_c} I\{|d_{ij}^{*(b)}| \leq \beta \bar{x}^*\} \geq 1 - \alpha. \quad (3)$$

This final step is carried out by ordering the systems in non-increasing order of the main output measure and working down the list until 3 is true for the set of systems at the top of the list but would not be true if the next system was included in the set.

5. Return  $S^*$ .

### 4.3 Python Implementation

An implementation of the optimization procedure, BootComp, was developed in Python 3.6.0. Core bootstrap code is written using standard Python libraries. Numpy 1.11.3, Pandas 0.19.2, and Matplotlib 2.0.0 were used for analysis. It is recommended that users download and install Anaconda that bundles Python 3.6 and its most popular Data Science extensions (<https://www.anaconda.com/download/>). A GitHub repository is available online (<https://github.com/CLAHRCWessex/BootComp>). Users unfamiliar with version control or GitHub can simply download the repository and de-compress the files. Details of the applied example are found in the Jupyter notebook `BootComp_WSC18.ipynb`.

## 5 NUMERICAL RESULTS

We present results for a real-world simulation model coming from healthcare. A relatively brief description of the context is provided below and we refer readers to Penn et al. (2018) for more details of the model and how it was used in practice.

### 5.1 Background to the simulation model

The discrete-event simulation model was used in a project with a UK National Health Service (NHS) clinical commissioning group, acute hospital and community health care provider in the South of England in 2017. The project investigated the design of an NHS community hospital rehabilitation ward in order minimize delays in the transfer of care of elderly in-patients from a large acute hospital. The new rehabilitation ward would be created from the merger of two geographically separate wards.

NHS England mandate single sex accommodation for patients. To accommodate this rule and provide some flexibility, beds within the wards could be put within single rooms or grouped in bays. Single rooms are in one sense the most efficient way to improve flow; however, single rooms are more expensive to safely staff. Bays of beds offer a good compromise to single rooms in terms of staffing costs and patient flow. An empty bay is assigned a gender corresponding to the gender of the first patient to be admitted. As patients are admitted and discharged from a ward, patients can be moved and the gender of the bay flipped to admit waiting patients. As there are no differences in process times for patients in a single room or within a bay, bays are less efficient for flow than single rooms. For example, a male patient requiring rehabilitation may not be able to be placed on a rehabilitation ward that has a free bed if the free bed is within a female designated bay. The simulation study, therefore, analyzed, the mix of single rooms and size/number of bays that were required to keep waiting times for admission low and costs acceptable.

The main primary output of the model are average waiting time for transfer to rehabilitation. Secondary outputs are utilization (occupancy) of beds, and the number of patient transfers between gender specific bays. The client requested a *set* of good systems as opposed to a single optimal number.

### 5.2 Experiments and comparisons

In stage one the model was used to conduct 1151 initial simulation experiments (stage 1 replications = 5; time unit = days; run length = 365 days). The replications can be found in three .csv files in the Git repository (`data\replications_wait_times.csv`; `data\replications_util.csv`; `data\replications_transfers.csv`).

Table 1 lists the parameters used in the two-stage optimization. In practice clients preferred an average utilization ( $\rho$ ) of at least 80% and the maximum number of patient transfers ( $t$ ) between bays was set to 50. The number of replications in stage 1 is based this on the pragmatic recommendations from Law and Kelton (2000); taking the upper bound of their recommended range of *at least 3-5 replications*. In the applied example, it took approximately 2.5 hours to generate the stage one replications for all 1151

systems, using commercial software and a machine of a similar specification that would be found in use in industry (Simul8 Professional 2017; Dell Laptop; 16GB RAM; i7 processor). Chernick (2007) warns of the limitations of the bootstrap for sample sizes below 10. We found that results were consistent if stage one replications were extended to 10 per system (with the associated 5 hour model run time) and consequently report the results with just 5 replications. The stage 1 and stage 2 values for  $\gamma$  were chosen to allow for the fact that errors will be higher in stage 1 due to the smaller number of replications, while  $\alpha$  was set to 0.05 to mimic accepted practice in ranking and selection procedures.

Table 1: Parameters used in the optimization.

Parameter	Stage 1	Stage 2
Utilization constraint	$\geq 80\%$	$\geq 80\%$
Patient transfers	$\leq 50$	$\leq 50$
Replications	5	50
$\gamma$	0.7	0.95
$\alpha$	0.05	0.05

### 5.3 Chance Constraints

Figure 1 illustrates an initial informal analysis of the conflict between the mean waiting time (Figure 1a), which is the main output measure, and the first chance constraint, which requires utilization to be greater than 80% (Figure 1b). For reasons of simplicity, the chart only includes results for systems that have exclusively single rooms and with the number of beds in the range 43-55. The asymptotic confidence intervals for mean utilization in Figure 1b suggest that ward sizes of 47 and lower have the highest chance of meeting the chance constraint. This would lead to a mean waiting time of between 5 and 30 hours.

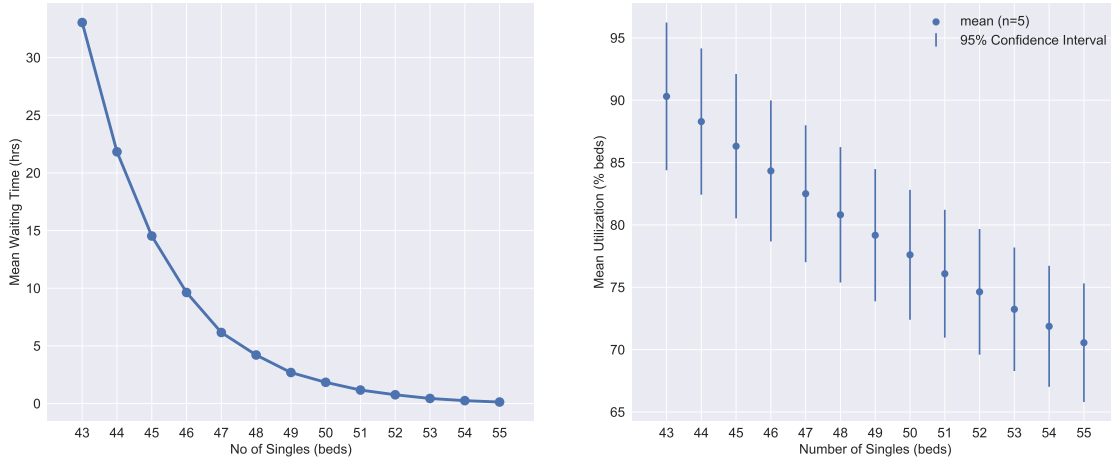


Figure 1: Illustration of stage 1 Utilization Chance Constraint.

### 5.4 Optimization Procedure

Figure 2 illustrates our optimization algorithm applied to the rehabilitation model. Stage 1 reduces the number of systems from 1151 to 34. Most of this reduction in systems is delivered by the chance constraints – reducing the feasible set to 177. A further 45 replications were conducted in stage 2, resulting in a total of 7285 individual runs of the model in stages 1 and 2. Stage 2 reduced the number of systems from 34

to 28. Table 2 lists the top 10 final elite systems. These all have 48 beds, and suggest that bay sizes of 3 or 4 are optimal.

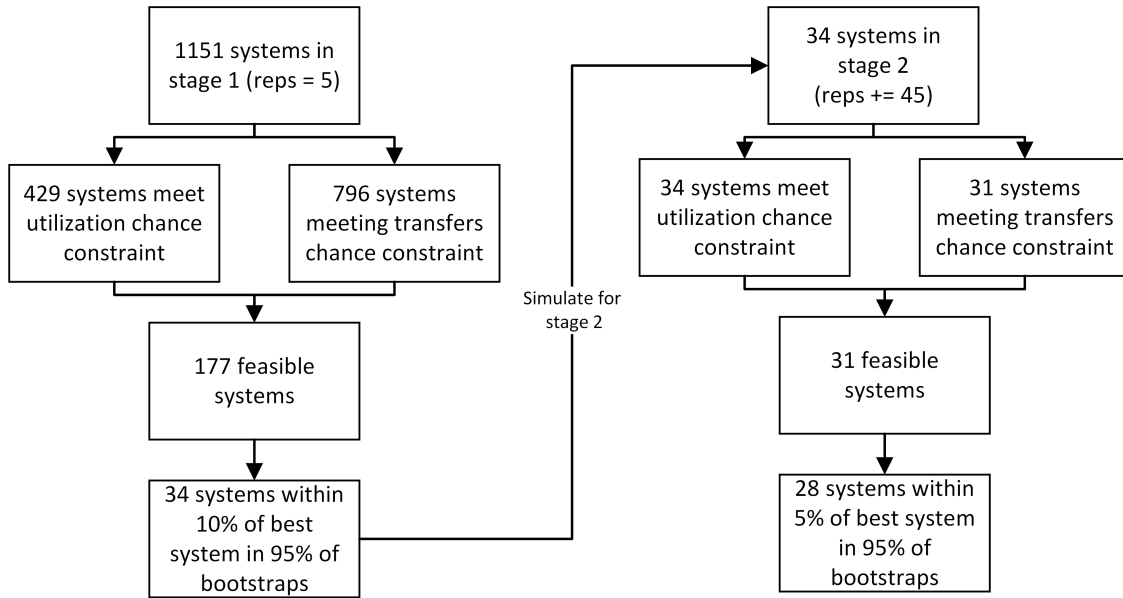


Figure 2: Optimization procedure in applied example

Table 2: Details of the top ten scenarios.

Total Beds	Bay Size	Number of Bays	Number of Singles	Waiting Time (days)	Utilization	Transfers
48	0	0	48	4.9	81.0	0.0
48	3	3	39	4.9	81.0	12.7
48	3	4	36	4.9	81.0	16.5
48	3	5	33	4.9	81.0	20.9
48	3	6	30	4.9	81.0	24.4
48	3	7	27	4.9	81.0	29.9
48	3	8	24	4.9	81.0	35.3
48	3	9	21	4.9	81.0	40.2
48	4	2	40	4.9	81.0	12.5
48	4	3	36	4.9	81.0	17.3

## 6 CONCLUSIONS AND FUTURE WORK

Our aim in this work was to develop a tool that is simple-to-use, fits within typical simulation practice, does not require the user to implement code to interact with the simulation models, and yet draws on recent research in ranking and selection. The result is a two-stage method, which reduces simulation effort by removing solutions that are likely to be infeasible or poorly performing at the end of the first stage, and goes on to return a subset of solutions that satisfy one or more chance constraints and are within a fixed tolerance of the best solution. We have demonstrated its use on a complex simulation from health care and this first set of experiments suggests that it works well at identifying good solutions.



In all cases we note that the method includes a number of parameters to be set by the user, including threshold probabilities for failing the chance constraints or for being within the given tolerance of the best solution. Users may wish to spend some time experimenting with the optimization parameters at each stage. In the applied example considered here, we found that increasing the value of  $\gamma$  from 0.7 to 0.8 in stage 1 led to a different set of elite solutions. More experimentation is needed to determine more clearly how the choice of these parameters affects the final solution.

We acknowledge that the results included here do not include a full comparison with other similar methods. It would be interesting to see how the computational effort and the solution quality compare with other subset selection ranking and selection algorithms from the literature and this is something that we will consider including in future work.

Our Python framework is free and open to simulation users now. No implementation is needed and it can be used with any study carrying out optimization via simulation (OvS). Nonetheless, testing of the framework is currently limited to a single empirical example. As a result, although preliminary results are promising, at this stage we cannot substantiate the reliability of our approach in practice. Future work will include testing the method on a wider variety of OvS problems.

## REFERENCES

- Andradóttir, S., and S.-H. Kim. 2010. “Fully sequential procedures for comparing constrained systems via simulation”. *Naval Research Logistics* 57:403–421.
- Bechhofer, R. 1954. “A single-sample multiple decision procedure for ranking means of normal populations with known variances”. *Annals of Mathematical Statistics* 25:16–39.
- Bekki, J. M., B. L. Nelson, and J. W. Fowler. 2010. “Bootstrapping-based fixed-width confidence intervals for ranking and selection”. In *Proceedings of the 2010 Winter Simulation Conference*, edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Huan, and E. Yücesan, 1024–1033. Piscataway, New Jersey: IEEE.
- Branke, J., S. E. Chick, and C. Schmidt. 2007. “Selecting a selection procedure”. *Management Science* 53:1916–1932.
- Chen, C.-H. 1996. “A lower bound for the correct subset-selection probability and its application to discrete event simulations”. *IEEE Transactions on Automatic Control* 41:1227–1231.
- Cheng, R. C. 2017. *Non-Standard Parametric Statistical Inference*. 1st ed. Oxford, UK: Oxford University Press.
- Cheng, R. C., and C. S. Currie. 2009. “Resampling Methods of Analysis in Simulation Studies”. In *Proceedings of the 2009 Winter Simulation Conference*, edited by M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, 45–59. Piscataway, New Jersey: IEEE.
- Chernick, M. R. 2007. *Bootstrap Methods: A Guide for Practitioners and Researchers*. New York: John Wiley & Sons, Inc.
- Chick, S. E., and K. Inoue. 2001. “New Two-Stage and Sequential Procedures for Selecting the Best Simulated System”. *Operations Research* 49:732–743.
- Chingcuanco, F., and C. Osorio. 2013. “A Procedure to Select the Best Subset Among Simulated Systems using Economic Opportunity Cost”. In *Proceedings of the 2013 Winter Simulation Conference*, edited by R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, 452–462. Piscataway, New Jersey: IEEE.
- Fu, M. C. 2015. *Handbook of Simulation Optimization*. 1st ed. New York, USA: Springer.
- Gao, S., and W. Chen. 2015. “A Note on the Subset Selection for Simulation Optimization”. In *Proceedings of the 2015 Winter Simulation Conference*, edited by L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, 3768–3776. Piscataway, New Jersey: IEEE.
- Gao, S., and W. Chen. 2016. “A new budget allocation framework for selecting top simulated designs”. *IIE Transactions* 48:855–863.
- Gupta, S. 1965. “On some multiple decision (selection and ranking) rules”. *Technometrics* 7:2255–245.

- Hong, L. J., J. Luo, and B. L. Nelson. 2015. "Chance constrained selection of the best". *INFORMS Journal on Computing* 27:317–334.
- Hunter, S. R., and R. Pasupathy. 2013. "Optimal Sampling Laws for Stochastically Constrained Simulation Optimization on Finite Sets". *INFORMS Journal on Computing* 25:527–542.
- Kim, S.-H., and B. L. Nelson. 2006a. "On the asymptotic validity of fully sequential selection procedures for steady-state simulation". *Operations Research* 54:475–488.
- Kim, S.-H., and B. L. Nelson. 2006b. "Selecting the best system". In *Elsevier Handbooks in Operations Research and Management Science: Simulation*, edited by S. G. Henderson and B. Nelson. Waltham, MA: Elsevier.
- Koenig, L., and A. Law. 1985. "A procedure for selecting a subset of size m containing the l best of k independent normal populations". *Communications in Statistics - Simulation and Computation* 14:719–734.
- Law, A. M., and W. D. Kelton. 2000. *Simulation Modeling & Analysis*. 3rd ed. New York: McGraw-Hill, Inc.
- Lee, L. H., E. P. Chew, S. Teng, and D. Goldsman. 2010. "Finding the Non-Dominated Pareto Set for Multi-Objective Simulation Models". *IIE Transactions* 42:656–674.
- Lee, S., and B. L. Nelson. 2014. "Bootstrap ranking and selection revisited". In *Proceedings of the 2014 Winter Simulation Conference*, edited by A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, 3857–3868. Piscataway, New Jersey: IEEE.
- Lee, S., and B. L. Nelson. 2015. "Computational improvements in bootstrap ranking and selection procedures via multiple comparison with the best". In *Proceedings of the 2015 Winter Simulation Conference*, edited by L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, 3758–3767. Piscataway, New Jersey: IEEE.
- Lee, S., and B. L. Nelson. 2016. "General-purpose ranking and selection for computer simulation". *IIE Transactions* 48:555–564.
- Nelson, B. L., and F. Matejcek. 1995. "Using common random numbers for indifference-zone selection and multiple comparisons in simulation". *Management Science* 41:1935–1945.
- Pasupathy, R., S. R. Hunter, N. A. Pujowidianto, L. H. Lee, and C.-H. Chen. 2014. "Stochastically constrained ranking and selection via SCORE". *ACM Trans. Model. Comput. Simul.* 25:1–26.
- Penn, M., T. Monks, A. Kazmierska, , and M. Alkoheji. 2018. "Designing and Redeveloping Generic Models in Healthcare." In *Proceedings of the OR Society Simulation Workshop 2018*, edited by A. Anagnostou, R. Meskarian, and D. Robertson. Stratford-Upon-Avon, UK: OR Society.
- Wang, Y., L. Luangkesorn, and L. J. Shuman. 2011. "Best-Subset Selection Procedure". In *Proceedings of the 2011 Winter Simulation Conference*, edited by S. Jain, R. R. Creasey, J. Himmelspace, K. P. White, and M. Fu, 4315–4323. Piscataway, New Jersey: IEEE.

## AUTHOR BIOGRAPHIES

**THOMAS MONKS** is the Director of Data Science at NIHR CLAHRC Wessex, University of Southampton. His research interest is applied simulation modelling, optimization and machine learning in healthcare. His email address is [thomas.monks@soton.ac.uk](mailto:thomas.monks@soton.ac.uk) and his ORCID is [orcid.org/0000-0003-2631-4481](https://orcid.org/0000-0003-2631-4481)

**CHRISTINE CURRIE** is Associate Professor of Operational Research in Mathematical Sciences at the University of Southampton, UK. Her email address is [Christine.Currie@soton.ac.uk](mailto:Christine.Currie@soton.ac.uk) and her ORCID is [orcid.org/0000-0002-7016-3652](https://orcid.org/0000-0002-7016-3652).