Optimized Decision Making About Library Shelf Space Through Simulation

Daina Bouquin, John DeBlase, James Topor

December 15, 2016

Abstract

Shelf space management is a common and ever-present challenge for libraries throughout the world. While many libraries have developed ad-hoc "rule of thumb" shelf space management protocols for their own internal use, the actual effectiveness of any such ad-hoc technique is, at best, difficult to gauge. This paper explores the use of Monte Carlo simulation for purposes of identifying the ideal amount of space to leave empty on library shelves such that the need to reshelve books due to library collection growth is minimized while also accommodating the largest possible quantity of books. To address this question, a Monte Carlo simulation that incorporates a wide variety of user-specified library management, collection size, and usage assumptions was developed using **R**. The simulation model indicates that for a small 1000-volume library collection housed within a total of fifty (50) thirty-six inch shelves, 5 inches of initial free shelf space will minimize the need for reshelving with very little associated risk of having to move any of the collection to off site storage. If zero risk of sending materials off site is required, 4.5 inches is the maximum recommended amount of initial free shelf space. While these results are specific to the simulated 1000-volume library, the simulation model can be applied to other library collections via simple adjustment of the relevant model parameters.

Key Words: Library Science, Shelf Shifting, Monte Carlo Simulation

Introduction

Shelf shifting is the process by which free space in a library's collection can be divided up and redistributed amongst shelves to distribute books more evenly throughout the collection. "Shifting is one of the most important steps necessary to ensure that a collection is maintained in proper order while allowing room for growth" [1]. The process of shelf shifting can be done on either a small or large scale whenever major changes occur within a library's physical collection. However, it can also be required when regular collection growth results in overfilling of shelves. While regular "weeding" and "de-duplicating" (periodic processes that remove out-of-date and/or rarely circulated volumes or duplicate volumes from library shelves) can themselves somewhat alleviate the need for periodic shelf shifting, such processes are usually not sufficient to prevent a collection's shelves from becoming overfilled. As a result, many libraries have adopted off-site storage models wherein a portion of the physical collection is stored remotely and is accessed via courier or other delivery service. However, the cost of off-site storage can be substantial, as is the cost of recalling material back to the main library from the remote storage facility. Similarly, shelf-shifting can be a time consuming and expensive process when shelves are allowed to fill over long periods of time. And while there has in the recent past been a notable shift toward electronic book collecting, physical collections are still maintained. It is therefore in a library's best interest to store as little offsite as possible and to simultanerously execute shelf shifting activities as little as possible.

With this context in mind, our research attempts to answer the following question through the use of Monte Carlo simulation:

- What is the ideal amount of space to leave empty on a book shelf to minimize the need to shelf shift while accomodating the largest possible quantity of books?

Implicit in this research question is the goal of limiting the amount of time needed to maintain the library's physical collection. For example, if a library can improve its ability to predict when a shelf shifting initiative will

likely be required based on variables such as the number of books periodically added, weeded, and de-duplicated within a collection, the library might not only be able to take steps to minimize the need for shelf shifting but also more effectively plan and budget for any required shelf shifts when they do occur.

Also implicit in this question is the assumption that the library in question has access to an off-site storage option. Granular metrics for libraries having such off-site storage (typically large academic libraries) are not publicly available, nor are metrics for their individual collections generalizable; different collections in different domains housed at different institutions are subject to very individualized collection development policies that impact the basic metrics one would use to simulate various processes associated with a library's physical collections. For these reasons, the herein described simulation is to be treated as a simple case study or "proof-of-concept" based on a simulated small collection from a single library having access to an off-site storage facility.

Background

The library to be used as a case example and from which metrics are drawn to create the initial set of simulation assumptions is the Wolbach Library at the Harvard-Smithsonian Center for Astrophysics in Cambridge, MA [1]. Metrics are gathered from Wolbach's Head Librarian who has access to Wolbach's in-development collection development and weeding policies, as well as circulation metrics associated with the physical collection.

The following facts are illustrative of Wolbach's practices and the benefits of using simulation to support decision making when it comes to the maintance of a physical library collection:

- **Shifting Time**: It takes one Wolbach staff person shifting at 10.87 inches of books per minute for 7 hours to shift a section of the Wolbach Collection. Due to the physical strain of this work, this sort of shift is typically split over 2 days. One can assume all staff members make at least $17/hour and that this timeframe does not take into account time needed for planning the shift. If it is necessary to shift many sections, this adds to the time significantly. Large shifts can take many weeks and get costly quickly.

- **Off Site Storage**: First time accession fees apply to all individual items stored off site. For instance, if Wolbach sent off site 108 items, a $1.26 charge per item would be applied resulting in a $136.08 charge. Each time an item is recalled to Wolbach, a courier is used with recall fees totalling $3.27 per item. Therefore, while complex decisions are made regarding what materials are sent off-site, the most substantial deciding factor is library shelf space. Turn around time for the delivery of material is typically one to two business days, but if material is needed immediately, "emergency" recalls can be made a price of $182 per trip. For these reasons it is advantageous to limit the amount of material stored off site whenever possible. For fiscal year 2017, Wolbach has budgeted $31.5K for off-site storage. There are currently over 10.7 million items at the off site Depository used by Wolbach, and Harvard Library is expanding its off-site storage to accomodate an even larger amount of material.

Literature Review

Our literature review focused on identifying applicable simulation strategies from the fields of library science and Monte Carlo simulation. Within the field of library science there are established methods and guidelines for shelf shifting [2] or adding shelving to libraries [3] and even for completely rearranging library spaces to accomodate more books [4]. However, estimating the amount of space to leave empty on a shelf for purposes of minimizing shelf shifts is somewhat imprecise. For example, documents and tools exist for estimating shelf space, time, and resources needed to shift a known collection [5] and there have been methods developed to inform decisions about shifting collections off site for the re-organization of a library [6]. In fact there are even commercial products available to help libraries with large collections move to a new building, for expanding existing shelving, for moving to entirely new locations, and for shifting books to more evenly distribute space within existing shelving [7]. However, there appear to be no specific tools available for estimating the ideal amount of shelf space to leave open for purposes of minimizing the need for shelf shifts and maximizing the number of volumes kept on site. There seem to be no known applications of simulation for these puposes either. Many libraries develop their own ad-hoc approaches to estimating collection growth rates (note: ad-hoc growth rates are generally only applicable

to periodical collections) or fall back on rules-of-thumb. For instance, there are numerous anecdotal references online to the general idea that one should leave at least 6 inches of space open on each shelf, but such anecdotes do not appear to be based on any type of mathematical or scientific analysis. Instead, it is common industry practice to leave space for at least one human hand width at each end of a shelf when shifting.

Monte Carlo simulation is a widely used method for solving both optimization and numerical integration problems through the use of random sampling. The general approach to problem solving through use of the method is well documented [8] as is the manner in which Monte Carlo simulation can be applied to various types of optimization problems [9], [10]. However, as was alluded to above, there appears to be no documented use of Monte Carlo simulation for purposes of minimizing the need for shelf shifting within a library while retaining the largest possible number of books on site. This apparent lack of prior application of Monte Carlo methods to such a problem suggests that the work described herein may, in fact, be novel.

Methodology

As stated in the **Introduction**, our research was focused on attempting to answer the following question:

- What is the ideal amount of space to leave empty on a book shelf to minimize the need to shelf shift while accomodating the largest possible quantity of books?

To address this question, a Monte Carlo simulation of a small 1,000 volume library collection was developed using **R**. The simulation model relies on a variety of assumptions and constraints, each of which is described below. The assumptions and constraints are based on generalized metrics gathered relative to Wolbach Library's physical collection, community, and practices. For purposes of building an easy to adapt proof-of-concept tool, the collection is substantially simplified. Each of the assumptions listed below can be adjusted to accomodate differing parameters within different library contexts.

Assumptions

- **Number of Books/Volumes**: The simulated library collection is comprised of a total of **1,000 books**. This represents a single, relatively small, collection within the broader library collection.

- **Types of Books/Volumes**: There are two types of books in the simulation: textbooks and non-textbooks. **5%** of the collection is assumed to be master copies of textbooks at the start of the simulation. Multiple copies of a textbook are typically housed within the library; e.g., multiple copies of new physical textbooks are routinely purchased for the small number of undergraduate Astronomy courses taught each semester at the Harvard-Smithsonian Center for Astrophysics, and duplication of textbooks is typical within any given subsection of the library. Within the simulation, the number of copies of a textbook is determined by randomly sampling from a uniform distribution bounded by **(1, 3)**. By contrast, non-textbooks are generally not duplicated within the library.

- **Book Widths**: The width of the simulated books varies between 1 and 2 inches according to a triangular approximation of a normal distribution. As such, when a book is added to the library a random sample from a triangular **(1, 2)** bound distribution is used to determine the width of that book. This distribution is based on architectual graphics standards used in library space design [11]. By defaulting to these standards the simulated library collection does not need to distinguish between paperback and hardcover volumes.

- **Shelves**: A total of **fifty (50) 36 inch shelves** are available for storage of the simulated library collection. Books are assigned to the simulated shelves in a serial manner at the start of the simulation until a user-specified minimum amount of free space remains on a given shelf. Once that minimum amount is achieved, shelving continues on the next available shelf. Any volumes not shelvable due to space constraints are sent "off-site" for the purposes of the simulation. The total number of shelves (50) represents the typical amount of shelving designated for a small subcollection within the Wolbach Library.

- **Books in Circulation**: We assume that **10%** of the total collection has been checked out for use by various library patrons prior to the start of the simulation. This value is a simplified assumption based on averages of monthly "COUNTER" statistics pulled from Harvard's **Cognos** reporting tool for Wolbach's non-periodical collection and could be modified for any alternative collection's circulation rate. Collection circulation rates tend to be seasonal in nature and are not typically tracked at the level of a subcollection. The width of any book that is in circulation is added back to the amount of free space available on the relevant shelf.

- **Percentage of Books Weeded**: Wolbach Library's general collection development policy is in the process of being reassessed. However, a general plan is in place to initiate a regular weeding policy. The simulation reflects that policy in that **2%** of the simulated collection is flagged for weeding during each round of the simulation. If a book is checked out at the time when weeding occurs, the book is considered exempt from weeding for that round of the simulation. This policy ensures that circulating books are not weeded from the collection.

- **Number of Textbooks De-Duplicated**: We determine the number of textbooks to be de-duplicated during each round of the simulation by sampling from a uniform **(1, 5)** bound distribution. A corresponding number of textbooks is then randomly selected from the simulated collection for de-duplication. During de-duplication, all copies of the selected textbooks are removed from the collection. However, a master copy is retained. If a copy of a textbook is flagged for de-duplication while it is in circulation, it is not immediately removed from the simulated collection but instead is flagged for removal when it is returned to the library.

- **Percentage of Books Checked In**: We assume that **20%** of books currently in circulation are returned to the library during each round of the simulation. Books to be checked back in are selected via simple random sampling of the books currently in circulation.

- **Percentage of Books Checked Out**: We assume that **2%** of books currently **not** in circulation are randomly selected for check out during each round of the simulation.

- **New Textbook Acquisitions**: We assume that **between 3 and 6** new textbooks and associated copies will be added to the collection during each round of the simulation, with the exact number determined via random sampling from a uniform **(3, 6)** bound distribution. The number of copies to be added to the master copy for each new textbook purchased is determined by sampling from a uniform (1,3) bound distribution. Non-full shelves are randomly selected for each new textbook and associated copies are colocated on the same shelf.

- **New Non-Textbook Acquisitions**: We assume that **between 5 and 26** new non-textbooks are added to the collection during each round of the simulation, with the exact number determined via random sampling from a uniform **(5, 26)** bound distribution. Random selection of non-full shelves is used to shelve the new non-textbooks.

- **Amount of Initial Free Shelf Space**: The simulation examines a series of initial free shelf space values ranging from **3 inches** up to **6 inches** in 0.5 inch increments. (*NOTE: Six inches was selected as the upper bound for the range to coincide with the "width of two human hands" free shelf space guideline espoused in many anecdotal library management discussions*). The amount of initial free shelf space is used during shelving of the virtual library collection at the start of each run of the simulation, with each of the 50 shelves guaranteed to have no less than the indicated amount of free space.

- **Shelf Filling**: A shelf is considered "full" if at least **95%** of its available space has been consumed. When a shelf surpasses 95% usage, no further new acquisitions may be added to that shelf. However, books from such a shelf that are in circulation when the 95% usage boundary is surpassed are returned to the shelf whenever they are returned to the library despite the shelf having been deemed "full" for other purposes.

- **Shelf Shift Required**: A shelf shift is assumed to be required whenever at least **5** shelves achieve "full" status.

Running the Simulation

The R-based simulation allows us to evaluate the effects of a variety of initial free shelf space values. Specifically, the R code implements and aggregates the results of 100 simulations for each of the initial free shelf space values of **3, 3.5, 4, 4.5, 5, 5.5**, and **6** inches. A single run of the simulation is deemed to be complete whenever at least 5 library shelves become "full" according to the **Shelf Filling** description given above. The results of the simulations for the various initial free shelf space values are then tabulated and compared to allow us to determine which of the initial free shelf space values allows for the smallest average number of shelf shifts for the given 1,000 volume collection while sending as few books as possible to off-site storage.

Overview of Simulation Logic

The simulation's API consists of a pair of dataframes (books and shelves) that represent the contents of the library and a set of methods that act upon those dataframes. The flowchart below outlines the tasks performed by those methods and the process flow for each simulation run.

```
                    ┌─────────────────────────────────────┐
                    │            Initialize               │
                    │   dataframes: books, shelves        │
                    │ counters: counter, full count,      │
                    │           deleted books             │
                    │ constants: reshelf threshold,       │
                    │           free space                │
                    └─────────────────────────────────────┘
                                    │
                                    ▼
                            ┌───────────────┐
                            │   weeding     │◄──────────────┐
                            └───────────────┘               │
  ┌──────────────────────────┐        │                     │
  │ purchasing texts & non   │        │    ( update shelves)│
  │ texts                    │        ▼                     │
  └──────────────────────────┘  ┌───────────────┐           │
              (Input)  ┄┄┄┄┄►   │ adding new    │           │
                                │   books       │           │
                                └───────────────┘           │
                                    │    ( update shelves)   │
                                    ▼                        │
                                ┌───────────────┐            │
                                │  check ins    │            │
                                └───────────────┘            │
                                    │    ( update shelves)   │
                                    ▼                        │
                                ┌───────────────┐            │
                                │  check outs   │            │
                                └───────────────┘            │
                                    │    ( update shelves)   │
                                    ▼                        │
                                ┌───────────────┐            │
                                │ deduplication │            │
                                └───────────────┘            │
                                    │    ( update shelves)   │
                                    ▼                        │
                              ◄ # full shelves >=     No   ┌──────────┐
                                reshelf threshold? ───────►│increment │
                              ◄                             │ counter  │
                                    │ Yes                  └──────────┘
                                    ▼
                            ┌───────────────┐
                            │    Return     │
                            │ counter, #    │
                            │ deleted books │
                            └───────────────┘
```

As shown in the flowchart, an initialization function is invoked at the start of each simulation. The initialization function is responsible for creating the virtual library that serves as the basis of the actual simulation run. At a high level, the initialization function can be thought of as creating a virtual collection of textbooks and non-textbooks and placing those books on virtual shelves subject to the constraints imposed by the book width, shelf count, shelf width, and shelf free space constraints described above in the **Assumptions** section. Any books that are non-shelvable due to space constraints are are considered to have been shipped to off-site library storage, and the number of books presumed to have been sent off site is tallied. Additionally, 10% of the virtual books are set to a status of "checked out", thereby indicating that they are in circulation prior to the start of the simulation run. Finally, a handful of constants and counters are initialized, including a constant "reshelf" threshold that triggers the termination of an individual simulation run. As mentioned above, a simulation is terminated whenever at least 5 shelves become "full". Other constants initialized include the the various "initial free shelf space" values mentioned above.

After initialization, the simulation enters a process loop, or "cycle", wherein the methods mentioned previously are applied to the virtual library. Each method applied is representative of a particular library process. Specifically, books are weeded from the collection; new textbooks and copies thereof are added to the shelves; new non-textbooks are added to the shelves; some number of books in circulation are returned to the library (i.e., "checked in") and reshelved; some number of books are checked out by library patrons; and some number of textbooks are selected for "de-duplication", wherein excess copies of outdated textbooks are removed from the collection. Each of these cycles is meant to approximate roughly one academic year's worth of shelf activity for the Wolbach Library. After each iteration of this cycle the virtual library shelves are examined to determine whether the "reshelf" threshold has been exceeded. If so, the simulation terminates and a count of the number of successful cycles is returned along with a count of the books that were sent to off site storage prior to the start of the simulation.

As was mentioned above, 100 such simulations are performed for each initial free shelf space value and the results are aggregated for subsequent analysis.

Model Verification & Validation

Of primary concern in the development of any simulation model is determining whether or not the behavior of the fabricated model is sufficiently representative of "real world" behavior to allow for its use as a practical decision-making tool. Objective model verification and validation generally requires the collection of fairly detailed metrics from the actual real world systems being modeled. In our case, while we were able to accummulate a variety of general metrics related to the operations of the Wolbach Library, due to both time and budgetary constraints we were unable to collect robust operational metrics sufficient to allow us to objectively verify and validate all aspects of the model described herein. Instead, we have employed a subjective model verification and validation approach that relies upon the subject matter expertise of a member of our team who is currently employed by the Wolbach Library (Daina Bouquin) and feedback from staff members at the Wolbach Library who assist in shelf-shifting activies. Through their expertise we were able to confirm the reasonableness of all model assumptions as well as the reasonableness of the model's output. Furthermore, their subject matter expertise has allowed us to achieve a high level of "face validity" for the model. As a result, we are confident that the model described herein can potentially be of use to others who are responsible for shelf space management at other libraries. However, the lack of availability of a truly objective model validation via comparison of the model's behavior to "real world" metrics represents a relevant model usage caveat. Furthermore, the generalizability of the simulation model may be somewhat limited due to the simplifying assumptions used as well as its status as a "proof-of-concept".

Results

How Soon Will A Shelf Shift Be Required?

Running the simulation 100 times for each of the previously defined initial free shelf space values allows us to approximate how soon the library will be required to engage in a shelf shifting effort for each of the respective initial free shelf space values. A summary of our findings for this time-based metric is provided in **Table 1**, where the **Inches** column represents the initial free shelf space value for the respective sets of 100 simulation runs and the **mu** column represents the average number of simulation cycles completed prior to at least 5 virtual library shelves become filled with books. As was mentioned above, each of the simulated cycles represents approximately

one academic year's worth of shelf activity for the Wolbach Library. The median, minimum, and maximum number of completed simulation cycles are shown in the **mdn**, **min**, and **max** columns, respectively, while the **sd** column represents the standard deviation for the number of completed simulation cycles. Finally, the **c_int** column represents the 95% confidence interval for the mean number of completed simulation cycles.

Table 1: Number of Iterations Until Shelf Shift Needed

Inches	mu	mdn	min	max	sd	c_int
3.0	4.77	4	1	14	2.411	(4.297, 5.243)
3.5	5.82	5	2	14	2.560	(5.318, 6.322)
4.0	7.05	7	2	19	3.141	(6.434, 7.666)
4.5	7.68	7	1	21	3.570	(6.980, 8.380)
5.0	9.17	8	3	17	3.172	(8.548, 9.792)
5.5	11.15	10	4	21	4.375	(10.293, 12.007)
6.0	11.17	11	3	28	4.120	(10.363, 11.977)

Given that each simulation cycle represents approximately one academic year of library shelf activity, **Table 1** shows that increasing the initial free shelf space value from 3 inches up to 6 inches more than doubles the amount of time the library might expect to pass before being required to engage in a shelf shifting effort. This result in and of itself is rather unsurprising since one would expect that a greater amount of initial free shelf space will delay the need for a shelf shift to a greater degree than will a relatively lesser amount of initial free shelf space.

Table 1 also clearly shows that the number of simulation cycles completed prior to at least 5 virtual library shelves becoming filled with books can vary greatly for any given initial free shelf space value due to the random variability inherent in the use and management of a library collection. For example, at 3 inches of initial free space we observe a minimum of 1 completed cycle and a maximum of 14. Similarly, at 4.5 inches of initial free space we observe a minimum of 1 completed cycle and a maximum of 21. However, the 95% confidence intervals shown in the rightmost column for each of the initial free shelf space values' 100 simulation runs indicate that, for practical purposes, the selection of any particular free shelf space value is likely to result in a delay in the need to shelf shift that is reasonably approximated by the respective **mu** values shown in the table.

Additionally, this first table also shows that increasing the initial free shelf space value from 4 inches to 4.5 inches does not appear to substantively increase the number of times the simulation cycle can be completed prior to the shelves becoming filled. However, increasing the value from 3 inches to 4 inches or from 4 inches to 5 inches does appear to substantively increase the number of times the simulation cycle can be completed prior to the shelves become filled. The boxplots shown below provide further evidence for this observation.

Results: # of Iterations Until Shelf Shift
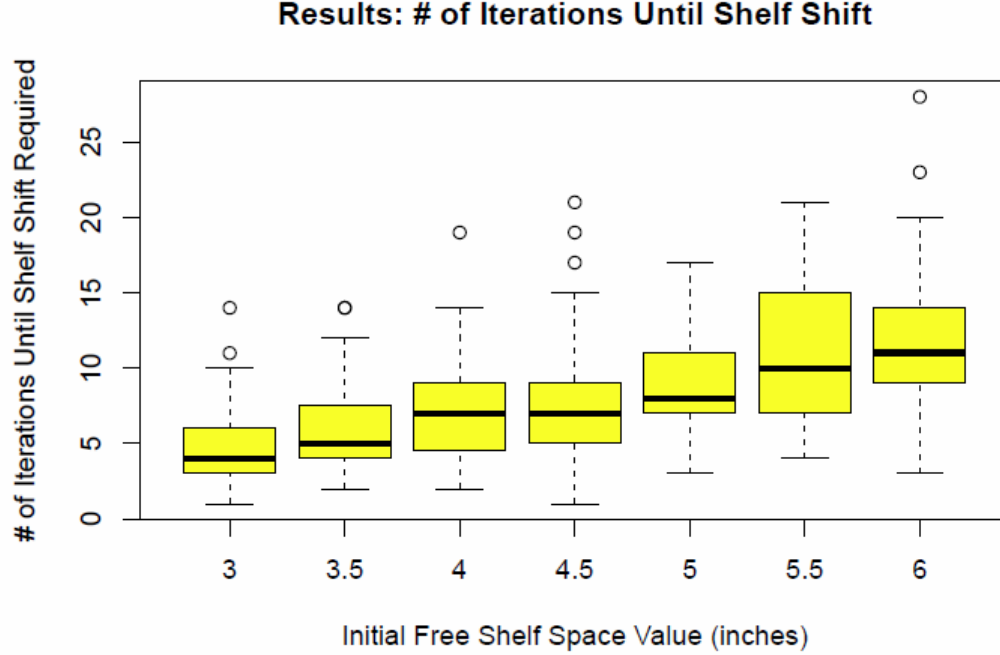
In fact, the boxplots also appear to demonstrate a substantive delay to the need for shelf shifting if the initial free shelf space value is increased beyond 5 inches by any 0.5 inch increment. However, as shown in **Table 2** below, this delay in the need for shelf shifting comes at the cost of having to send an ever increasing number of books to offsite storage.

How Many Books Must Be Shipped to Off Site Storage?

The simulation also allows us to evaluate whether an increase in the amount of initial free shelf space will result in an increase in the need for off site storage for some portion of the library's collection. **Table 2** summarizes our findings regarding the need for offsite storage. The structure of **Table 2** is analogous to that of **Table 1** with the variable being analyzed in **Table 2** representing the average number of books shipped offsite due to lack of shelf space for each respective set of 100 simulation runs.

Table 2: Number of Books Sent Off Site

Inches	mu	mdn	min	max	sd	c_int
3.0	0.00	0	0	0	0.000	(0.000, 0.000)
3.5	0.00	0	0	0	0.000	(0.000, 0.000)
4.0	0.00	0	0	0	0.000	(0.000, 0.000)
4.5	0.00	0	0	0	0.000	(0.000, 0.000)
5.0	0.15	0	0	7	0.857	(-0.018, 0.318)
5.5	8.87	9	0	24	5.733	(7.746, 9.994)
6.0	25.15	25	10	37	5.511	(24.070, 26.230)

Table 2 clearly shows that the library should expect no increase in the number of books stored offsite unless the initial free shelf space value is set to at least 5 inches. At 5 inches, off site storage may be required for a very small number of books (mu = 0.15), though a 5 inch space would allow the library to extend the average amount of time between required shelf shifts from the approximately 7.68 cycles achievable at 4.5 inches to 9.17 cycles (see Table 1). Increasing the initial free space value beyond 5 inches results in a substantive increase in the number of books that must be sent off site due to the fact that imposing ever-larger free space constraints will necessarily reduce the amount of shelf space available for books. The figure shown below clearly summarizes this phenomena based on the results shown in **Table 2**.

Results: Avg. # of Books Sent Off Site
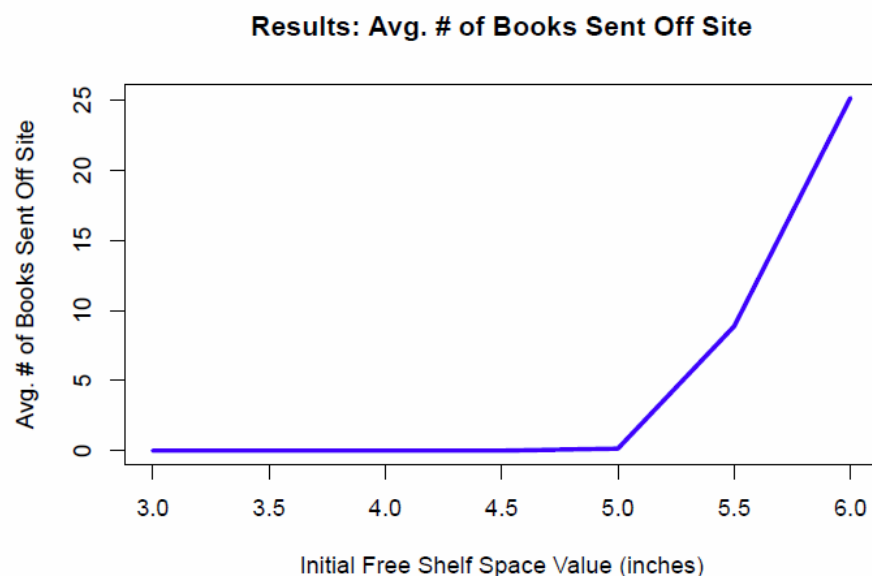
As such, it seems reasonable to conclude that for this small simulated library collection it would be advantageous to require 5 inches of initial free space on the shelves. Doing so both minimizes shifting activities and leaves the library with little risk of having to send materials from the collection off site. If the library does not have access to either an off-site storage option or the use of additional shelving, 4.5 inches would be the maximum allowable amount of initial free shelf space.

Summary and Future Works

The purpose of our research was to attempt to apply Monte Carlo techniques to the problem of minimizing shelf shifting activities in libraries while maximizing the number of volumes that can be kept on the library's shelves. The resulting model has a high degree of face validity and serves as a foundation for a flexible protocol that could be adapted to suit any library subcollection as long as metrics supportive of the assumptions and parameters outlined herein are available. The approach we've outlined also provides guidance to librarians relative to which metrics must be collected in order to take advantage of this new tool. Additionally, the model we've described here could be used to help a library improve its ability to predict when shelf shifting will be required or to enable proactive steps that could minimize the need for shelf shifts. In fact, the model could also help a library plan and budget for any required shelf shifts in a more effective manner. Finally, since the model is **R**-based (and therefore "open"), it could be made accessible to even the most budget-constrained libraries.

It should be noted that the conclusions described herein apply *only* to a small, simplified subcollection of the Wolbach Library. Future work could include extending the simulation model to allow it to be applied to larger and more complex library collections. For example, the simulation described here modeled only 1/60th (i.e., 1000 volumes) of the Wolbach's Library's physical collection, whereas most library collections are typically much larger in size. Furthermore, the scale of our simulated collection was necessarily constrained due to both a lack of readily available library metrics relative to a number of simulation model variables and the throughput limitations of the software and hardware we used.

Future work could also include augmenting the current model to reflect public library collections where textbooks are not a substantial portion of the collection, and extending the model to allow for seasonal fluctuations in the circulation metrics of a library collection. Similarly, the model could be extended to account for projected decreases in physical book aquisitions due to shifts toward ebook purchases to more realistically reflect the contemporary complexities of modeling a physical library collection.

References

[1] http://library.cfa.harvard.edu/

[2] https://www.libraries.rutgers.edu/rul/staff/access_serv/coll_mgt/overview_of_shifting.pdf

[3] https://scholarsbank.uoregon.edu/xmlui/bitstream/handle/1794/12208/Moving+Library+Collections_+UO+Libraries2.pdf?sequence=1

[4] http://www.tandfonline.com/doi/abs/10.1300/J105v05n01_07

[5] https://osc.hul.harvard.edu/liblab/projects/collection-shift-estimation-visualization-tool

[6] http://www.tandfonline.com/doi/abs/10.1080/01462679.2016.1212755

[7] http://www.aautomate.com/products/shelf-space-manager

[8] Paxton, Pamela, et al. "Monte Carlo experiments: Design and implementation." Structural Equation Modeling 8.2 (2001): 287-312.

[9] Homem-de-Mello, Tito, and Guzin Bayraksan. "Monte Carlo sampling-based methods for stochastic optimization." Surveys in Operations Research and Management Science 19.1 (2014): 56-85.

[10] Li, S.P.. "A Guided Monte Carlo Method for Optimization Problems." International Journal of Modern Physics C 13.10 (2002): 1365-1374

[11] https://www.graphicstandards.com/

_____

Appendix

This Appendix contains the research results, source R code, and associated relevant output from our final writeup and our simulation efforts. Additional supplemental graphics and detailed explanations of our simulation efforts are also included.

Code Appendix

setup.R

The setup method initializes and returns the books and shelves dataframes to the parent scope.

```r
setup = function(Nvols, Nshelves, shelf_width, sfree_space, textb_masters){

    library(triangle)

    # set max shelf space allowed to be consumed by books at start of simulation
    # sfree_space represents the total amount of free space in INCHES
    max_shelved <- shelf_width - sfree_space

    # Initialize 'books' data frame
    books <- data.frame(book_id = 1:Nvols, btype = character(Nvols), copym = numeric(Nvols),
                        dupeof = numeric(Nvols), shelf_id = numeric(Nvols), width = numeric(Nvols),
                        chkdout =numeric(Nvols), dedupe = numeric(Nvols), stringsAsFactors = FALSE)

    # set book widths: distrib is between 1-2 inches; assume UNIFORM distribution
    # now sample from triangular distribution: Number of samples = number of books ('Nvols')
    a <- 1 ; b <- 2 # upper/lower bound
    books$width <- rtriangle(Nvols, a, b, (a + b)/b )

    books$btype <- 'n' # initialize book type to n for non-textbook

    # then set percentage of books to textbook to textbook master copies
    # start by picking random sample of books
    tbm <- sort(sample(books$book_id, (Nvols * textb_masters), replace = FALSE))

    # For each book_id in tbm, set btype = 't', copym = 1, and create copies of
    for (i in tbm) {
        # if the book has not already been set to type 't', proceed
        # otherwise ignore and continue
        if (books$btype[i] != 't') {
            # set btype to 't'
            books$btype[i] <- 't'
            # set copym to '1'
            books$copym[i] <- 1
            # get number of copies of textbook via random uniform sample between 1 and 3
            tb_copies <- round(runif(1, min = 1, max = 3))

            # create copies of textbook directly adjacent to master copy
            k <- i + 1
            while (k <= Nvols & (k <= i + tb_copies) ) {
                books$btype[k] <- 't'
                books$dupeof[k] <- books$book_id[i]
                # set width of copy to width of master
                books$width[k] <- books$width[i]
```

```r
                k <- k + 1
            }
        }
    }

    # Initialize 'shelves' data frame  --> full 0 or 1 flag for whether shelf is full
    shelves <- data.frame(shelf_id = 1:Nshelves, in_use = numeric(Nshelves),
                          perc_used = numeric(Nshelves), full=numeric(Nshelves), stringsAsFactors = FALSE

    k <- 1
    # sum book widths to ensure they don't exceed (max shelf width - free_space)
    for (i in 1: Nshelves) {
        # while space used on shelf < max space consumed by books and book index < Nvols
        while ((shelves$in_use[i] + books$width[k]) < max_shelved & (k <= Nvols) ) {
            # assign shelf to next book
            books$shelf_id[k] <- shelves$shelf_id[i]
            # add width of book to total used on current shelf
            shelves$in_use[i] <- shelves$in_use[i] + books$width[k]
            k <- k + 1
        }
    }

    #############################
    # remove all books that were not able to be shelved
    books <- books[which(books$shelf_id != 0),]

    # calculate number of deleted books
    delbks <- Nvols - nrow(books)

    # select 10% of book_id values to set to 'checked out' status
    c_out <- sample(books$book_id, (nrow(books) * .10), replace = FALSE)

    # For each book_id in c_out, set chkdout = 1 and subtract
    # book width from shelf space used for appropriate shelf_id
    for (i in c_out) {
        # set chkdout flag to 1
        books$chkdout[i] <- 1
        # subtract width of chkdout book from space in use on its assigned shelf
        shelves$in_use[books$shelf_id[i]] <- shelves$in_use[books$shelf_id[i]] - books$width[i]
    }
    shelves$perc_used = shelves$in_use/shelf_width  # sets perc_used

    return(list(books=books, shelves=shelves, delbks = delbks)) # return books shelves

}
```

methods.R

The public API that acts on the books and shelves dataframes.

```r
# Utility functions

# helper function that duplicates textbooks and assigns ids
duplicate_textbooks = function(last_indx, texts){
```

```r
        texts$book_id = seq(last_indx+1, last_indx+nrow(texts))  # id and set to master
        texts$copym = 1

        for(i in 1:nrow(texts)){
            n_dups = runif(1,1,3)
            dups = data.frame(lapply(texts, function(j){
                rep(j[i],n_dups)
            }))
            dups$copym = 0                    # unset master, set btype, dupeof id
            dups$btype = 't'
            dups$dupeof = dups$book_id
            texts = rbind(texts,dups)          # bind to purchase order
        }
        texts$book_id = seq(last_indx+1, last_indx+nrow(texts))     # reset book ids
        return(texts)
}


book_widths = function(n,a,b){        # utility returns book widths between a and b for n books
    rtriangle(n, a, b, (a + b)/b )
}

get_shelf_ids = function(shelves){     # return vector of shelf ids that are available for shelving
    shelves$shelf_id[which(!shelves$full)]
}

# Simulation processes API

# updates shelves dataframe at end of each round
update_shelves = function(books, shelves, shelf_width){

    for (i in 1:nrow(shelves)){
        new_width = sum(books[which(books$shelf_id==i & !books$chkdout),]$width) # calculate new width
        shelves[shelves$shelf_id==i,]$in_use = new_width       # set new widths for each shelf
    }
    shelves$perc_used = shelves$in_use/shelf_width  # reset perc_used
    shelves$full[which(shelves$perc_used > 0.95)] = 1  # set flag at full greater than 95%
    # if shelf was full and books are removed(deduping or checkout) set back to zero
    shelves$full[which(shelves$perc_used <= 0.95 & shelves$full == 1)] = 0

    return(shelves)
}

# set of purchases - 5 to 25 books (simulated book widths)
# set of textbook purchases (simulated book widths) - 2 copies of 3 to 5 books
# set flag to 'n', 't' for text-book or non-textbook
# return a new dataframe to be appended to books in outer scope
# this is a blank order... all field's build/add logic is done in the add()/update_shelves() functions
purchase_books = function(flag, min=5, max=26){

    if (flag=='n')      # branch for text/nontext -- R throws an exception if not 'n /'t'
        n_books = round(runif(1,min,max))
    else if (flag=='t')
        n_books = round(runif(1,min,max))
```

```r
    width = book_widths(n_books,1,2) # calls utility function book_widths()
    btype = rep(flag,n_books)        # sets flags

    book_id = copym = dupeof = shelf_id = chkdout = dedupe = rep(0,n_books)
    purchase_order = data.frame(
        book_id, btype,copym,dupeof,shelf_id,width,chkdout,dedupe
    )

    return(purchase_order)
}

# special logic for appending a textbook df to main library df
# textbooks need to be shelved together
add_textbooks = function(books,new_books, shelves){

    last_indx = tail(books$book_id,1)
    new_books$shelf_id = sample(get_shelf_ids(shelves),nrow(new_books)) # shelf ids assigned before dupli
    new_books = duplicate_textbooks(last_indx,new_books)
    books = rbind(books,new_books)
    return(books)
}



#speical logic for appending a non-textbook df to main library df
add_non_textbooks = function(books,new_books, shelves){

    last_indx = tail(books$book_id,1)
    new_books$book_id = seq(last_indx+1, last_indx+nrow(new_books))
    new_books$shelf_id = sample(get_shelf_ids(shelves),nrow(new_books)) # assign shelf id from available
    books = rbind(books,new_books)
    return(books)
}



# round of weeding - 2% of the collection -> DO NOT weed books that have been checked out
weeding = function(books, perc=0.02){
    n_to_pull = round(nrow(books) * perc) # number of rows to randomly pull
    books = books[-sample(which(!books$chkdout), n_to_pull), ] # pull em, update and return
    # renumber indices of rows after removal of books
    rownames(books) <- 1:nrow(books)
    return(books)
}



# round of de-duplicating (textbooks) - follows logic in email. Last step in process
de_dup = function(books){

    # n number of textbooks to dedupe
    n_dedup = round(runif(1,1,5))
    # get book ids for the master copies to be deduped
    master_ids = books[sample(which(books$copym==1),n_dedup),]$book_id

    # set dedupe flags for any duplicates with chkdout=1; pull the ones with chkdout=0
```

```r
    if(length(books[which(books$dupeof %in% master_ids & books$chkdout==1), ]$dedupe > 0))
        books[which(books$dupeof %in% master_ids & books$chkdout==1), ]$dedupe = 1

    # if there is a subset of duplicates that are not checked out remove them
    if(nrow(books[which(books$dupeof %in% master_ids & books$chkdout==0), ])!=0)
        books = books[-which(books$dupeof %in% master_ids & !books$chkdout), ]

    # renumber indices of rows after removal of books
    rownames(books) <- 1:nrow(books)

    return(books)
}

# 2% of the collection gets newly checked out -> check flag
# need to set deduping flag for textbooks that get checked out <<<<--
check_outs = function(books, perc=0.02){
    n_to_chkout = round(nrow(books) * perc) # number of rows to randomly checkout
    book_chkout = books[sample(which(!books$chkdout), n_to_chkout), ]$book_id

    books$chkdout[books$book_id %in% book_chkout] = 1
    return(books)
}


# 20% of the collection gets checked in (not the same as the check outs)
check_ins = function(books, perc=0.2){
    n_to_pull = round(length(books$chkdout[books$chkdout == 1])* perc) # 20% of previously checked out bo

    book_returns = books[sample(which(books$chkdout==1), n_to_pull), ]

    # check_in deduping logic... if flag set to 1 in batch of returned books, remove from dataframe
    if (any(book_returns$dedupe==1)){
        dedupe_rows = book_returns[book_returns$dedupe == 1,]
        books = books[-which(books$book_id %in% dedupe_rows$book_id), ]
        # renumber indices of rows after removal of books
        rownames(books) <- 1:nrow(books)
    }

    books$chkdout[books$book_id %in% book_returns$book_id] = 0
    return(books)
}
```

single_sim.R

Performs a single round of simulation by implementing the setup and API methods. The application logic is outlined in the above flowchart.

```r
# Single sim wrapped in function
source("setup.R")    # these need to be in the same folder...
source("methods.R")

single_sim = function(nvols=1000, nshelves=50, shelf_width=36, sfree_space=3,
                      textb_masters=0.05, reshelf_thresh=5, seed_val= NULL){
```

```r
    set.seed(seed_val)    # specify seed
    data = setup(nvols, nshelves, shelf_width, sfree_space, textb_masters) # returns list, need to set to

    #init books and shelves
    books = data$books
    shelves = data$shelves
    delbks <- data$delbks

    counter = 0
    full_count = 0
    reshelf_thresh = 5

    while(T){
        ### SIM CODE
        books = weeding(books,0.02)
        # need to update available shelf space after weeding
        shelves= update_shelves(books,shelves,36)

        # purchase books - functions create data frames containing new books
        # to be added to collection
        new_texts = purchase_books('t',3,6)
        new_ntexts = purchase_books('n',5,26)

        # append data frame containing new_ntexts to books data frame
        books = add_non_textbooks(books,new_ntexts,shelves) # needs logic for checking shelf availability
        shelves= update_shelves(books,shelves,36)

        # append data frame containing new_texts to books data frame
        books = add_textbooks(books,new_texts,shelves) # needs logic for checking shelf availability
        shelves= update_shelves(books,shelves,36)

        books = check_ins(books,0.2) # needs logic for checking shelf availability
        shelves= update_shelves(books,shelves,36)

        books = check_outs(books,0.02)
        shelves= update_shelves(books,shelves,36)

        books = de_dup(books)
        shelves= update_shelves(books,shelves,36)
        ####

        full_count = sum(shelves$full)        # get number full
        if(full_count >= reshelf_thresh)      # break if >= threshold
            break
        counter = counter + 1
    }

    # clear memory of data, books, shelves
    rm(data, books, shelves)

    # The counter value is number of rounds
    # return(counter)
    return(list(iters = counter, delbks = delbks))
}
```

main.R

The main simulation code. This script soruces the single_sim file, runs 100 simulations on an increasing amount of inital shelf space and returns the results.

NOTE: A parallelized version of main also exists in the repo that substantially cuts down on run time. A non-parallelized version is presented here for clarity.

```r
library(knitr)

set.seed(123)

setwd("c:/users/hammer/Documents/github/Sim_Project")

source("single_sim.R")  # import single_sim...

# set the sequence for the required amounts of free shelf space
free_inches <- seq(3, 6, 0.5)

df_dim <- length(free_inches)

# create data frame to store results of separate simulations for different free shelf space values
res_master <- data.frame(Inches = free_inches, mu = numeric(df_dim), mdn = numeric(df_dim),
                         min = numeric(df_dim), max = numeric(df_dim), sd = numeric(df_dim),
                         c_int = character(df_dim), stringsAsFactors = FALSE)

# create data frame to store count of books moved to offsite storage
os_master <- data.frame(Inches = free_inches, mu = numeric(df_dim), mdn = numeric(df_dim),
                        min = numeric(df_dim), max = numeric(df_dim), sd = numeric(df_dim),
                        c_int = character(df_dim), stringsAsFactors = FALSE )

# create a vector to store results of 100 sims of one free shelf space value
sim_results = numeric(100)
del_bks = numeric(100)

# for each item in the free shelf space sequence, run the full simulation
# and collect results
df_ind <- 1

# create a data frame to store all simulation results for use outside of the for loop
bxp_df <- data.frame(matrix(, nrow=100, ncol=0))
delbks_df <- data.frame(matrix(, nrow=100, ncol=0))

for (k in free_inches) {
  # run 100 iterations for each free shelf space value
  for(i in 1:100){
    res_list <- single_sim(sfree_space = k)
    sim_results[i] <- res_list$iters
    del_bks[i] <- res_list$delbks
  }

  # add col containing sim results to boxplot dataframe
  bxp_df <- cbind(bxp_df, sim_results)
  # set the name of the new col to the number of inches of free space
  colnames(bxp_df)[colnames(bxp_df) == 'sim_results'] <- toString(k)
```

18

```r
  # add col containing del_bks to deleted books dataframe
  delbks_df <- cbind(delbks_df, del_bks)
  # set the name of the new col to the number of inches of free space
  colnames(delbks_df)[colnames(delbks_df) == 'del_bks'] <- toString(k)

  # display results for free shelf space = k
  print(sprintf("Free Space = %1.1f inches", k))
  print(summary(sim_results))
  xl <- sprintf("Free Space = %1.1f inches", k)
  hist(sim_results, breaks = 20, col = 'yellow', xlab = xl)

  # create a boxplot of the results for free shelf space = k
  # boxplot(sim_results, main = xl, col = 'yellow' )

  # store iteration results in master results data frame
  res_master$mu[df_ind] <- mean(sim_results)
  res_master$mdn[df_ind] <- median(sim_results)
  res_master$min[df_ind] <- min(sim_results)
  res_master$max[df_ind] <- max(sim_results)
  res_master$sd[df_ind] <- round(sd(sim_results), 3)
  # calculate 95% confidence interval: First we need SE = sd/sqrt(N)
  SE <- sd(sim_results) / sqrt(100)
  lowerB <- mean(sim_results) - 1.96 * SE
  upperB <- mean(sim_results) + 1.96 * SE
  res_master$c_int[df_ind] <- toString(sprintf("(%2.3f, %2.3f)", lowerB, upperB))

  # store iteration results in master results data frame
  os_master$mu[df_ind] <- mean(del_bks)
  os_master$mdn[df_ind] <- median(del_bks)
  os_master$min[df_ind] <- min(del_bks)
  os_master$max[df_ind] <- max(del_bks)
  os_master$sd[df_ind] <- round(sd(del_bks), 3)
  # calculate 95% confidence interval: First we need SE = sd/sqrt(N)
  SE <- sd(del_bks) / sqrt(100)
  lowerB <- mean(del_bks) - 1.96 * SE
  upperB <- mean(del_bks) + 1.96 * SE
  os_master$c_int[df_ind] <- toString(sprintf("(%2.3f, %2.3f)", lowerB, upperB))

  df_ind <- df_ind + 1
}

# side-by-side boxplots of number of iterations until shelf shift needed
boxplot(bxp_df, main = "Results: # of Iterations Until Shelf Shift", xlab = "Initial Free Shelf Space Val

kable(res_master, caption = "Number of Iterations Until Shelf Shift Needed")

# linear plot avg number of books sent off site
plot(x = os_master$Inches, y = os_master$mu, type = "line", col = "blue", lwd = 3,
     xlab = "Initial Free Shelf Space Value (inches)",
     ylab = "Avg. # of Books Sent Off Site",
     main = "Results: Avg. # of Books Sent Off Site")

kable(os_master, caption = "Number of Books Sent Off Site")
```