LivSim: An Open-Source Simulation Software Platform for Community Research and Development for Liver Allocation Policies

Vikram Kilambi[1,2,3*] MS PhD, Kevin Bui[1,2] MS, Sanjay Mehrotra[1,2,4] PhD

[1]Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL

[2]Center for Engineering and Health, Institute for Public Health and Medicine, Northwestern University Feinberg School of Medicine, Chicago, IL

[3]RAND Corporation, Boston Office, Boston, MA

[4]Northwestern University Transplant Outcomes Research Collaborative (NUTORC), Comprehensive Transplant Center, Northwestern University Feinberg School of Medicine, Chicago, IL

***Corresponding Author**

Vikram Kilambi, MS PhD

Associate Operations Researcher

20 Park Plaza Suite 920

Boston, MA 02116

Phone: 617-338-2059 x8635

Email: vkilambi@rand.org

Authorship Information:

Vikram Kilambi[1*] MS PhD, Kevin Bui[2] MS, Sanjay Mehrotra[3] PhD

*Equal contribution credit

[1]Developed software, debugged software, and edited the article.

[2]Developed software, debugged software, and drafted the article.

[3]Supervised software, reviewed software, and edited the article.

Authors have no conflicts of interest to disclose.

This work is funded by National Science Foundation award CMMI-1131568 and Agency for Healthcare Research and Quality award 1R36HS024840-01.

Abbreviations:

DSA- Donor Service Area

LSAM – Liver Simulated Allocation Model

MELD – Model for End-Stage Liver Disease

OPTN - Organ Procurement and Transplant Network

SRTR - Scientific Registry of Transplant Recipients

UNOS - United Network for Organ Sharing

We developed an open-source software called LivSim, an alternative to the Liver Simulated Allocation Model (LSAM v Aug 2014)[1] created by the Scientific Registry of Transplant Recipients (SRTR). LivSim is designed to perform discrete-event simulation of liver allocation in the United States' Organ Procurement and Transportation Network (OPTN). Written in Python 3.4.2, the software is available for download at https://github.com/LivSim2017/LivSim-Codes.

As the standard simulation tool in transplantation, LSAM assesses alternative liver allocation policies. LSAM allows changes to only liver allocation rules, model parameters, and boost points for testing policies. However, its code cannot be modified when a user tests a policy that is beyond its capabilities. For example, at the time when optimized[2] and concentric[3] neighborhoods were being developed, they could not be tested on LSAM because of their complex geographic structures for sharing. LSAM's inflexibility motivated LivSim's development.

Unlike LSAM, LivSim operates primarily at the DSA level. Using LivSim, the user can directly specify how procured organs are shared between DSAs. Therefore, LivSim was able to examine the allocation performances of the 8-district structure[4], optimized neighborhoods[2], and the concentric-neighborhoods solution[3].

LivSim operates in a similar manner as does LSAM, and its statistical results are approximate to LSAM's results[2]. LivSim maintains lists of transplant candidates, recipients, and donors; initializes a starting waitlist; and reads in LSAM's input streams of patient arrivals, organ arrivals, status updates/progressions of waitlist candidates, and patient re-listings. Each waitlist candidate is assigned a Model for End-Stage Liver Disease (MELD) score, ABO blood type, Status 1 exception, and HCC exception.

During a status progression, LivSim updates the candidate's MELD score and potentially removes the candidate from the waitlist or indicates the candidate's death. When a donated liver arrives, LivSim creates a list of potential, blood-type-compatible candidates prioritized by their MELD scores and their proximity to the DSA that procure the organ. Iterating through the list, LivSim uses LSAM's organ acceptance model[1] to determine whether the candidate accepts the offered liver for transplant. After transplant, LivSim will determine whether the transplant recipient will relist and, if necessary, return the recipient to the waitlist after calculating the time until graft failure. The LivSim architecture is summarized in Figure 1.

When the simulation finishes, it outputs the following statistics:

- DSA-average MELD at transplant and standard deviation
- DSA-median MELD at transplant and standard deviation
- Number of transplants by year and by DSA
- Number of waitlist mortalities by year and by DSA
- Number of waitlist removals by year and by DSA
- Average transplant waiting time by year and by DSA
- Number of procured organs sent or received between DSAs

Using the output statistics, LivSim can generate statistics on post-transplant mortalities, re-listing and re-transplant mortalities, and transport distances and times.

Being open-source, the LivSim's code can be directly modified to incorporate new features, update survival and acceptance models, change allocation rules, etc. The availability and transparency of LivSim encourages transplantation researchers to strive for creative allocation policies that follow robust design principles[5] in potentially saving more lives and reducing geographic disparity overall.

Figure Legends

Figure 1: LivSim's flow chart describing the simulation process from start to finish.

References

1. *Liver Simulated Allocation Model: User's Guide* [computer program]. SRTR; 2014.
2. Kilambi V, Mehrotra S. Improving Liver Allocation Using Optimized Neighborhoods. *Transplantation.* Feb 2017;101(2):350-359.
3. Mehrotra S, Kilambi V, Bui K, et al. A Concentric Neighborhoods Solution to Disparity in Liver Access that contains current UNOS Districts. *Transplantation.* 2017.
4. Gentry SE, Massie AB, Cheek SW, et al. Addressing geographic disparities in liver transplantation through redistricting. *American journal of transplantation : official journal of the American Society of Transplantation and the American Society of Transplant Surgeons.* Aug 2013;13(8):2052-2058.
5. Mehrotra S, Kilambi V, Gilroy R, et al. Modeling the allocation system: principles for robust design before restructuring. *Transplantation*. 2015;99:278-281.