JURNAL EMACS
[Engineering, MAthematics and Computer Science] Vol.6 No.3 September 2024: 201-260 e-ISSN: 2686-2573 DOI: 10.21512/emacsjournal.v6i3.12344

Simulation Techniques in Sugarcane Transportation Model Using R Programming Language

I Gusti Agung Anom Yudistira[1*], Asysta Amalia Pasaribu[2], Aryusmar[3]

[1,2] Statistics Department, School of Computer Science,
[3] Language Center, Computer Science Department, Faculty of Humanities,
Bina Nusantara University,
Jakarta, Indonesia 11480
i.yudistira@binus.ac.id; asysta.amalia@binus.ac.id;
aryusmar@binus.ac.id

*Correspondence: i.yudistira@binus.ac.id

Abstract *– The R programming language is widely recognized for its capabilities in Monte Carlo simulations and numerical analysis. The objective of this paper is to create an R script that simulates sugar transportation in a circular flow (Field-Factory-Field). Therefore, this study aims to leverage R for simulating discrete events, focusing specifically on transportation systems, with an emphasis on sugar transportation in a circular flow model. This script's design is significant, as there is limited research on simulating circular systems. It will explore various system parameters, including: 1) the total volume of sugar successfully transported during the simulation; 2) the duration sugar remains within the transportation system; 3) The length of queues for various transportation modes at both the field and factory; 4) the number of trips completed by each transportation mode; and 5) the utility level of all resources available within the system. A key focus of this research is the creation of a script for circular simulation in sugar transportation, which can subsequently be expanded into a more intricate system. The research process consists of the following stages: 1) conducting a literature review, 2) outlining the sugar transportation system, 3) defining assumptions and system boundaries, 4) conceptually designing the transportation system, 5) developing the programming code, 6) testing and verifying the model, 7) validating the model, and 8) performing experiments on the model. The results of the analysis demonstrate that the open-source programming language R is a powerful tool for modeling the sugar transportation system.*

Keywords: *R Programming; Simulation; Circular Transportation System; Sugarcane*

I. INTRODUCTION

The transportation of raw materials in the agro-industry, particularly agricultural commodities, is often subject to perishability and necessitates careful handling to maintain yield levels, which in turn impacts the productivity and quality of factory output. Sugarcane, a key agro-industrial product, faces similar challenges, especially in its transportation. The quality of sugarcane deteriorates significantly due to delays in milling, which is more detrimental than the chemical or mechanical degradation that occurs later in the processing stage. Djamaris (1984) notes that sugar loss from delays in milling can result from extended storage in the fields, transportation delays, or holdups during unloading at the factory. Consequently, the transportation of sugarcane from fields to factories is a critical issue for companies aiming to enhance production efficiency (Bantacut, T. et al., 2012).

This transportation system involves three key activities: 1. Harvesting and loading in the field, 2. Transporting sugarcane from the field to the factory and back, and 3. Unloading at the factory. These processes will be represented through discrete event simulation using the R programming language. Similar to Python, R is a free programming tool, but it offers superior visualization capabilities. Originally, R was designed for Monte Carlo simulation and numerical computation, making it particularly effective for these applications.

R's advanced features for statistical analysis and data visualization make it an excellent choice for this research. Furthermore, the author's experience with R adds to this justification (Kabacoff, R. 2022; Rahmany, M., et al. 2020). Since 2017, R has included the simmer package, which allows for the creation of discrete event simulation models.

Khamidatullailiyah et al. (2020) conducted a simulation study on the Cutting, Loading, and Transport (TMA) of sugarcane using AnyLogic software, with transportation not being the primary focus of their research. This study aims to create a discrete event simulation model specifically for sugarcane transportation, employing the simmer package in R. A significant challenge in this transportation model is the development of R scripts for discrete event simulation, particularly for a circular route that goes from the field to the factory and back to the field. The authors have found limited literature addressing the creation of R scripts for closed discrete event simulations of this nature. The data utilized in this study are hypothetical, generated by computer simulations, and will be elaborated upon in the Methodology section. The findings of this research can serve as a valuable reference for other researchers looking to use R as an alternative tool for developing simulation models.

II. METHODS

The This research employs a literature review methodology, utilizing both descriptive and exploratory approaches. The research process consists of the following steps:

- *Literature Review*: The initial phase involves thoroughly understanding the functions provided by the simmer package and the organization of the simulation environment within the simmer class.
- *Chart or Flowchart Development*: A chart or flowchart is created to enhance comprehension of the system and to aid in translating it into R code. The diagram illustrating the simulation entity's path will be included in the discussion section (see Figure 1).

2.1 Creating Hypothetical Data and R Script

Once the charts and trajectories have been effectively designed, the next step involves developing the R script, a crucial element of this research. The algorithms and R scripts will be detailed in the discussion chapter. The hypothetical data, which plays a vital role in the R script, acts as input for the simulation. This includes: a) the mode of transportation utilized, which is trucks, categorized into two types: medium trucks (SD), with a capacity ranging from 5 to 10 tons, and large trucks (BS), with a capacity of up to 20 tons. The two types of trucks have a total of 3 and 6 units, respectively. b) In the field, there are three loading spots designated for sugarcane processing: one for the Large Truck and two for the Medium Truck. c) The loading time for sugarcane for the Medium Truck follows a uniform distribution ranging from 45 to 75 minutes, while the loading time for the Large Truck follows a uniform distribution from 60 to 150 minutes. The choice of probability distribution and its parameters can be modified based on field observations. d) The travel time for the Medium Truck from the field to the factory is determined by a uniform distribution between 20 and 45 minutes, whereas the travel time for the Large Truck is determined by a uniform distribution between 30 and 50 minutes. e) The carrying capacity of the Medium Truck is defined by a uniform distribution between 9 and 10 tons. while for the Large Truck it is generated from a uniform distribution (13.5, 15) tons; f) The unloading process is carried out at the factory and only one unit (carrier) is provided; g) The unloading time for the Medium Truck is generated from a triangular distribution.

The Medium Truck's travel time is modeled using a triangular distribution with parameters (3.5, 8, 4.5) minutes, while the Large Truck's travel time follows a triangular distribution of (7, 15, 8.5) minutes. Additionally, the travel time for the Medium Truck from the Factory to the Plantation is derived from a uniform distribution ranging from 10 to 30 minutes, whereas the Large Truck's travel time is based on a uniform distribution from 20 to 30 minutes. Running the simulation and validating the interactive R model allows for a step-by-step verification of the system's logic. To achieve more consistent simulation results and to calculate the standard error of the generated statistics, the simulation is conducted up to 100 times.

The analysis of the simulation outputs reveals three primary tables:

- *Arrival*: This table logs the simulation time along with the duration of activities for the entities (trucks).
- *Resources*: This table tracks server utilization during the sugarcane loading and unloading processes, the status of the truck queue, and the number of trucks in the loading and unloading system.
- *Attributes*: This table records the volume of sugarcane being loaded, currently in transit, and successfully unloaded at the factory.

These three tables facilitate the generation of statistical summaries, which include the average number of trips, queue lengths at both the field and factory, and the total volume of sugarcane transported by each truck over a span of 1,200 minutes. The accompanying graphs feature a Step Plot illustrating the queue at the field and factory, a Step Plot depicting server utilization, and a histogram representing the volume of sugarcane.

This paper outlines the process of modeling and simulation, beginning with problem formulation and concluding with decision-making. The focus of this research is specifically on model development, particularly the creation of simulation models. However, it does not cover data collection, testing model assumptions, or conducting experiments on the model.

2.2 System Overview

The sugarcane transportation system consists of three main subsystems: the loading process of sugarcane in the field, the transportation from the field to the factory (round trip), and the loading and unloading process for milling at the factory before returning to the field. For the purposes of this study, the harvesting subsystem is excluded to simplify the analysis, although the loading of sugarcane in the field remains a component of the system. The transportation of sugarcane from the field to the factory is carried out using three large trucks and six medium-sized trucks, each designated for different transportation routes. The factory features a single unloading area equipped with a rake.

2.3 Loading of Tied Sugarcane

The manual loading of tied sugarcane onto transport vehicles varies in duration based on the truck size. Medium trucks typically require between 45 and 75 minutes for loading, while large trucks take longer, with a loading time ranging from 60 to 150 minutes.

2.4 Journey from Plantation to Factory

The travel time for trucks is considered to be uniformly distributed. For medium-sized trucks, the travel time from the plantation to the factory ranges from 20 to 45 minutes. In contrast, large trucks have a longer travel time, taking between 30 and 50 minutes.

2.5 Unloading Cargo at the Factory

The primary tool used for unloading bundled sugarcane is the rake. This implement features a long iron rod with a claw-shaped end, designed to rake the sugarcane and pull it onto the cane carrier. In the unloading process at the factory, large trucks take precedence. Both large and medium trucks form a queue to access this loading and unloading service, but only one unloading spot is available. The unloading duration is expected to follow a triangular distribution, differing between the two truck types. For medium trucks, the unloading time ranges from a minimum of 3.5 minutes to a maximum of 8 minutes, with an average of 4.5 minutes. In contrast, large trucks have an unloading duration that varies from a minimum of 7 minutes to a maximum of 15 minutes, averaging 8.5 minutes.

2.6 Factory-Garden Journey Process

Once the sugarcane unloading is finished, the truck promptly heads back to the Garden. It returns empty. The return travel time is assumed to follow a uniform distribution. For a medium truck, the return journey to the Garden takes between 20 and 30 minutes, while a large truck typically takes about 30 to 40 minutes. Upon arrival at the Garden, the loading process resumes as usual. This cycle continues uninterrupted for 20 hours each day (1,200 minutes).

III. RESULTS AND DISCUSSION

The transportation process is depicted in Figure 1 below.

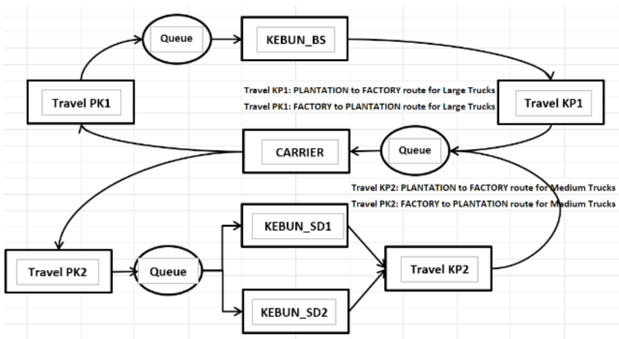

Figure 1. Illustration of the Sugarcane Transportation System

The sugarcane transportation system is modeled using R programming along with the simmer package. The simulation begins by establishing an environment called "sugarcane," followed by the definition of ten input parameters: MUAT_SD, MUAT_BS, BONGKAR_SD, BONGKAR_BS,Travel_KPSD, Travel_KPBS, Travel_PKSD, Travel_PKBS, Beban_BS, and Beban_SD. MUAT_SD and MUAT_BS indicate the loading durations for Medium Trucks (SD) and Large Trucks (BS), respectively. BONGKAR_SD and BONGKAR_BS denote the unloading durations for Medium and Large Trucks. Additionally, Travel_KPSD and Travel_KPBS represent the travel times for Medium and Large Trucks from the plantation to the factory.

Travel_PKSD and Travel_PKBS represent the travel times for Medium and Large Trucks from the Factory to the Plantation. Beban_SD and Beban_BS denote the volumes of sugarcane (measured in tons) transported by these trucks.

The loading area for Large Trucks consists of a single unit, referred to as KEBUN_BS in Figure 1.

In contrast, the loading areas for Medium Trucks include two units, designated as KEBUN_SD1 and KEBUN_SD2. There is one unloading area at the factory, marked in Figure 1 with a box labeled CARRIER. Consequently, this transportation system comprises three resources: KEBUN_BS (with a capacity of 1), KEBUN_SD1 (with a capacity of 2), and CARRIER (with a capacity of 1).

In this simulation model, a total of ten entities are created, including a dummy entity named "dummy," which serves to set the global variable 'Total' to a value of 0. The dummy entity will follow a path designated as traj00. The model includes three large trucks, represented by the entities TrukBSA, TrukBSB, and TrukBSC, each assigned a priority value of 1 (priority=1). TrukBSA will navigate along the trajectory TrajBSA, while TrukBSB and TrukBSC will follow the paths TrajBSB and TrajBSC, respectively. Additionally, six medium trucks are represented by the entities TruckSDA, TruckSDB, TruckSDC, TruckSDD, TruckSDE, and TruckSDF. Each of these entities will enter their corresponding trajectories: TrajSDA, TrajSDB, TrajSDC, TrajSDD, TrajSDE, and TrajSDF. All nine trucks are assumed to be available at the start of the process, at time point 0.

3.1 Simulation Algorithm

1) Define a function in R called `Trans_Tebu`. This function should accept inputs for generating the objects `MUAT_SD`, `MUAT_BS`, and `Beban_SD`, which have been previously outlined. Additionally, it should take an argument `rn`, which defaults to 1200 minutes, indicating that the simulation will terminate after this duration. The `Trans_Tebu` function will include the following commands.

2) Set up a discrete event simulation environment using the `simmer` package and assign it the name `tebu`. The corresponding R command for this is `tebu<-simmer("tebu")`.

3) Define the entity trajectory, which encompasses the commands for size, timeout, and release. The size function indicates that the entity (in this instance, the Truck) occupies the server when it is available but moves to the queue subsystem when the server is occupied. The timeout function tracks the duration the entity utilizes the server, while the release function signifies that the entity has relinquished the server, thereby making it available again. A total of 12 trajectories are established, including: a) kebunBS, which represents the trajectory for the queue of Large Trucks at the Garden. Once the Large Truck completes its time in the queue subsystem at the garden server, it advances to the Garden-Factory travel subsystem, with the timeout function recording the travel duration; b) kebunSD, which is analogous to the kebunBS trajectory but is designated for Medium Trucks. c) The factory serves as the path for the queue subsystem at the factory server, which is subsequently followed by the Factory-Garden travel subsystem. The duration of this journey is tracked using the timeout function. d) traj00 is a placeholder trajectory used to initialize the global variable Total to zero tons at the start time. e) trajBSA, trajBSB, and trajBSC are trajectories that merge the plantationBS trajectory with the factory trajectory for Trucks A, B, and C, respectively. These trajectories illustrate the trucks' journeys from the plantation to the factory and back. At the conclusion of these trajectories, an activation function is included to prompt the entities to return from the plantation to the factory. f) trajSDA, trajSDB, trajSDC, trajSDD, trajSDE, and trajSDF are trajectories that integrate the plantationSD trajectory with the factory trajectory for Medium Trucks A, B, C, D, E, and F, respectively. An activation function is also appended at the end of these trajectories.

4) Incorporating additional resources (servers) into the sugarcane simulation environment. The functions used are add resource (KEBUN_BS, 1), add resource (KEBUN_SD, 2), and add resource (CARRIER, 1). The KEBUN_BS server, designated for loading sugarcane in the field for large trucks, has a capacity of 1 unit. The KEBUN_SD server, intended for loading sugarcane for medium trucks, has a capacity of 2 units. Lastly, the CARRIER server serves as the unloading area for sugarcane at the factory, with a capacity of 1 unit.

5) Entity Generation. The R function used for creating entities is `add_generator`, which takes three arguments: the entity name, the trajectory the entity will follow, and the initial time at which the entity enters the trajectory or system. For example, the command `add_generator ("dummy", traj00, at(0), Mon=2)` creates an entity called "dummy," which enters the trajectory `traj00` at time 0. This action triggers the command `set_global("Total", 0)`, setting the global variable `Total` to 0. Similarly, Truck A, represented by the entity "TrukBSA," is generated using the function `add_generator("TrukBSA", trajBSA, at (0), mon=2, priority=1)`, allowing "TrukBSA" to enter the trajectory `trajBSA` at the start of the simulation. Trucks B and C, along with Medium Trucks A, B, C, D, E, and F, are generated in the same manner.

6) The sugar transportation system under investigation operates over a time span of 1,200 minutes. To ensure that the simulation does not exceed this duration, the function R is executed with the argument (1,200). In the Trans_ Tebu function, the value of 1,200 is specified as the input parameter rn = 1,200.

7) The simulation output is produced by three R functions: get_mon_arrivals, get_mon_ resources, and get_mon_ attributes. These functions generate the tables labeled Arrivals, Resources, and Attributes (refer to Appendix 1).

8) The Trans_Tebu function will be executed and repeated 50 times using the replicate function. The command is structured as follows: dt_rep <- replicate (n, Trans_ Tebu(…), simplify = FALSE) # dt_rep will hold the results of the replications.

The complete code/script can be accessed at the link Trans_Tebu.R.txt. Additionally, the R-format program code is available in the attached link.

This simulation model generates three outputs, presented as datasets (tibble class objects). The first output is the Arrivals dataset, which consists of six variables: name, start time, end time, activity time, resource, and replication. The name variable identifies the entities in the system, start time indicates when an entity enters the system, end time marks when it exits, activity time reflects the duration of the entity's activity within the system, resource denotes the name of the utilized resource, and replication refers to the specific simulation run. Consequently, the Arrivals dataset effectively captures the activities of entities within the system.

The second dataset, known as the Resources dataset, comprises eight variables: resource, time, server, queue, capacity, queue size, system, limit, and replication. The resource variable identifies the name of the resource utilized, while time denotes the specific moment in the simulation. The server variable indicates the number of entities being served, and queue reflects the number of entities waiting in line. Capacity represents the maximum capacity of the resources, and queue size indicates the maximum permissible length of the queue, which is assumed to be infinite in this study. The system variable refers to the total number of entities within the system, and limit signifies the cap on the number of entities present, which is not restricted in this analysis. Lastly, replication pertains to the repetition of the simulation. Consequently, the output from the Resources dataset illustrates the phenomena occurring within the resources (KEBUN_BS, KEBUN_SD, and CARRIER).

The third output consists of the Attributes dataset, which includes five variables: time, name, key, value, and replication. The time variable indicates the point in the simulation, while name refers to the entity's designation. The key represents the attribute's name utilized in the model, value denotes the attribute's corresponding value, and replication indicates the specific simulation run. The primary focus of this output is the attribute labeled Total, which reflects the weight of the sugarcane bonds successfully delivered to the factory. An example of this output is available in the Appendix. Additionally, the simulation output generated using R can be further analyzed to produce statistical summaries, visualizations, and estimates of system parameters.

IV. CONCLUSION

The R programming language, which originally excelled in Monte Carlo simulations, has now proven to be a valuable tool for modeling discrete event simulations, such as those found in sugar transportation systems. The accompanying program code is structured for clarity and aligns with the flowchart we developed (Figure 1). In this model, entities (Large Trucks and Medium Trucks) circulate in a loop from the Field to the Factory and back to the Field, with the script crafted in R. The results of the analysis can be monitored by reviewing the output generated by the log function, which is implemented at each point of change in the simulation status (refer to the log function in the complete R script Trans_Tebu.R.txt). The simulation results can be further analyzed, leveraging R's robust capabilities in statistical analysis. This study utilizes hypothetical data, which could be enhanced in future research by incorporating real data. Additionally, the system's scope can be broadened to encompass the sugarcane harvesting process and to incorporate break schedules if needed. The simmer package is well-suited to handle the complexities of such systems, although it may necessitate more intricate coding.

ACKNOWLEDGMENTS

This research was carried out with the resources made available by Bina Nusantara University, and the author is sincerely grateful for this support. Colleagues in the Statistics and Mathematics Department have been invaluable discussion partners, contributing significantly to the success of this research. We extend our appreciation to our friends at Bina Nusantara University for their encouragement and assistance throughout this project.

REFERENCES

Bantacut, T., & Sukardi, I. A. S. (2012). Sugar Loss in the Loading and Transport Cutting System at the Sindang Laut and Tersana Baru Sugar Mills in Cirebon. Journal of Agricultural Industrial Technology, 22(2), 115-121.

De Santis, A., Giovannelli, T., Lucidi, S., Messedaglia, M., & Roma, M. (2023). A simulation-based optimization approach for the calibration of a discrete event simulation model of an emergency department. Annals of Operations Research, 320(2), 727-756.

Ebert, A., Mengersen, K., Ruggeri, F., & Wu, P. (2021). Curve Registration of Functional Data for Approximate Bayesian Computation. Stats, 4(3), 762-775.

Kabacoff, R. (2022). *R in action: data analysis and graphics with R and Tidyverse*. Simon and Schuster.

Khamidatullailiyah, Y. G. N., Yaqin, M. A., & Utomo, A. H. (2020). Simulasi Model Proses Tebang Muat Angkut (TMA) On Farm pada Tanaman Tebu. *Jurnal Teknologi Informasi dan Terapan*, *7*(2), 118-124.

Rahmany, M., Zin, A. M., & Sundararajan, E. A. (2020). Comparing tools provided by python and r for exploratory data analysis. *IJISCS (International Journal of Information System and Computer Science)*, *4*(3), 131-142.

Smeets, B., & Ucar, I. (2020). Introduction to simmer. (https://r-simmer.org/articles/simmer-01-introduction.html)

Stark, M. (2020). Why data visualization is important. (www.analytiks.com)

Tyler, J. M., Murch, B. J., Vasilakis, C., & Wood, R. M. (2023). Improving uptake of simulation in healthcare: User-driven development of an open-source tool for modelling patient flow. Journal of Simulation, 17(6), 765-782.

Ucar, I., Smeets, B., & Azcorra, A. (2017). Simmer: discrete-event simulation for R. arXiv preprint arXiv:1705.09746.

Yudistira, I. G. A. (2021). Development of discrete event simulation based on the simmer package in R. Engineering, Mathematics and Computer Science Journal (EMACS), 3(2), 79-85.

ATTACHMENT

1. R coding in R format
   Link: 02-Trans_Tebu.R
2. R coding in txt format
   Link: Trans_Tebu.R.txt
3. Example of simulation output

Table I. Arrivals Dataset

```
> Data_out
$Arrivals
# A tibble: 96 × 6
   name     start_time end_time activity_time resource replication
   <chr>         <dbl>    <dbl>         <dbl> <chr>          <int>
 1 TrukSDA0        0       68.7         68.7  KEBUN_SD           1
 2 TrukSDB0        0       72.3         72.3  KEBUN_SD           1
 3 TrukSDA0       92.5     96.6          4.02 CARRIER            1
 4 TrukSDB0       95.0    102.           5.13 CARRIER            1
 5 TrukBSA0        0      110.         110.   KEBUN_BS           1
 6 TrukSDD0        0      138.          65.9  KEBUN_SD           1
 7 TrukSDC0        0      139.          69.9  KEBUN_SD           1
 8 TrukBSA0      146.     155.           8.46 CARRIER            1
 9 TrukSDC0      170.     175.           5.59 CARRIER            1
10 TrukSDD0      177.     184.           6.63 CARRIER            1
# ℹ 86 more rows
# ℹ Use `print(n = ...)` to see more rows
```

Table II. Resources Dataset

```
$Resources
# A tibble: 199 × 9
   resource  time server queue capacity queue_size system limit replication
   <chr>    <dbl>  <int> <int>    <dbl>      <dbl>  <int> <dbl>       <int>
 1 KEBUN_BS   0        1     0        1        Inf      1   Inf           1
 2 KEBUN_BS   0        1     1        1        Inf      2   Inf           1
 3 KEBUN_BS   0        1     2        1        Inf      3   Inf           1
 4 KEBUN_SD   0        1     0        2        Inf      1   Inf           1
 5 KEBUN_SD   0        2     0        2        Inf      2   Inf           1
 6 KEBUN_SD   0        2     1        2        Inf      3   Inf           1
 7 KEBUN_SD   0        2     2        2        Inf      4   Inf           1
 8 KEBUN_SD   0        2     3        2        Inf      5   Inf           1
 9 KEBUN_SD   0        2     4        2        Inf      6   Inf           1
10 KEBUN_SD  68.7      2     3        2        Inf      5   Inf           1
# ℹ 189 more rows
# ℹ Use `print(n = ...)` to see more rows
```

Table III. Attributes Dataset

```
$Attributes
# A tibble: 241 × 5
    time name       key       value replication
   <dbl> <chr>      <chr>     <dbl>       <int>
 1   0   ""         Total      0              1
 2  68.7 "TrukSDA0" Beban      9.31           1
 3  68.7 "TrukSDA0" BONGKAR    4.02           1
 4  68.7 "TrukSDA0" TRAVEL_PK 23.7            1
 5  72.3 "TrukSDB0" Beban      9.36           1
 6  72.3 "TrukSDB0" BONGKAR    5.13           1
 7  72.3 "TrukSDB0" TRAVEL_PK 25.9            1
 8  96.6 ""         Total      9.31           1
 9  96.6 "TrukSDA0" Beban      0              1
10 102.  ""         Total     18.7            1
# ℹ 231 more rows
# ℹ Use `print(n = ...)` to see more rows
```