ISSN 2347 - 3983
Volume 8. No. 10, October 2020
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter238102020.pdf
https://doi.org/10.30534/ijeter/2020/238102020

Discrete Event Simulation of Production and Logistics Processes for the industrial environment using R Programming

D.Phanindra Kshatra[1], K. Dileep[2], Anil Kumar. M[3], Sk. Khadar Basha[4], R Venkatesh[5]
[1]Koneru Lakshmaiah Education Foundation, India, phanindra@kluniversity.in
[2]Koneru Lakshmaiah Education Foundation, India, Dileep.kotte@gmail.com
[3]Koneru Lakshmaiah Education Foundation, India, anilmtcr@gmail.com
[4]Koneru Lakshmaiah Education Foundation, India, Bashazeeshan786@gmail.com
[5]Koneru Lakshmaiah Education Foundation, India, rentalavenkatesh@gmail.com

ABSTRACT

This paper aims to apply and evaluate the production scenario using Discrete Event Simulation (DES) through R programming, The study conducted here is on a production flow line where the goal is to apply the programming for system and evaluation of simulation flow in the manufacturing environment to identify the bottleneck, buffer capacity and the behavior of the system. The system is modelled, and methods for manufacturing environment are reviewed using R programming. The challenges and requirements of DES using R for the future are highlighted.

Key words: Modelling, Simulation, Discrete Event Simulation (DES), Computer Simulation, R Programming, R simmer

1. INTRODUCTION

For efficient management of a manufacturing system without causing any change to the present working conditions implementing virtual manufacturing environment is essential [1]. The production scenario is in constant pressure for process efficiency for improving their production performance and simultaneously eliminating the bottlenecks to meet customers' demands in time [1].
Simulation has become the core, and advanced methodology that is used in the production environment, simulation for a system will benefit for optimization and validate the performance of various processes in the system, it also helps to detect bottlenecks and eliminate inefficiencies before the start of production itself [2]. Experiments through computer simulations with Programming like R is a cost-effective, ease of use and time can be efficiently utilized to get results faster [3]. R programming has the excellent capability for statistics, visualization and customizable graphics, R has a remarkable ability and flexibility for simulation and analysis through simulation code which can be implemented from scratch, in this paper, we have discussed a case study of a production line with some ten machines and will illustrate the R programming capabilities for simulation and analysis [4],[5].

2. CASE STUDY

This case study is about the manufacturing of a product where raw materials enter into the production system to station -1 and will go to any of the free machine (priority rules are not set in this case study), once the process complete in Station-1, the semi-finished material will pass on to station-2 and further to station-3. If all the machines in any of the three stations are occupied with material during process, the raw material which is to be processed will be in Queue waiting for its turn to be processed [4],[5]. Once the processes are completed, the finished will get out of the system.

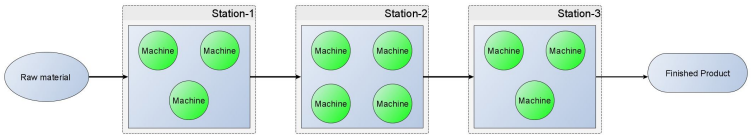

Figure 1: Production System

The flow of products in the system can visualize as follows

```
> # plot the Model
> plot(product)
> plot(product, verbose = T)
```

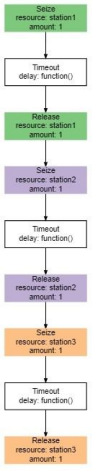

Figure 2: Flow of raw material in the system

One can observe that the raw material first Seize station 1 [4],[5]. The raw material movement is observed with the help of normal distribution with the use of Timeout function, this function mimics the real-time system, and after the estimated Timeout the station-1 releases the semi-finished raw material and further down the simulation continues with the next process which is inline, and the final product leaves the system [6], [7].

3. PRODUCTION SYSTEM PROGRAM IN R

3.1 LIBRARY

The first step in the process is to load the library of simmer which is used for Discrete event simulations and also simmer .plot for graphical visualization [4],[5].

```
> rm(list=ls()) ## Reset Environment
> cat("\014") ## clear console
> library(simmer)
> library(simmer.plot)
```

3.2 SYSTEM ENVIRONMENT

The next step is to create an environment for the production system and one can see that the system is empty because ther e are no resources are present in the system [8], [9].

```
> # Create Production System Environment
> sys<- simmer("Production System")
> sys
simmer environment: Production System | n
ow: 0 | next:
```

3.3 ADDING RESOURCES

With the help of piping operator (%>%) which is a chain of functions together where the output of one function is the input to another, so resources are added using the function add_resource to the system, and three stations are created, and each station have machines inside them [8],[9].

```
> sys %>%
+   add_resource("station1",3)%>%
+   add_resource("station2",4)%>%
+   add_resource("station3",3)
```

One can see that the system has three stations, and at this cur rent situation, there is no movement in the system.

```
simmer environment: Production System | n
ow: 0 | next:
{ Monitor: in memory }
{ Resource: station1 | monitored: TRUE |
server status: 0(3) | queue status: 0(Inf
) }
{ Resource: station2 | monitored: TRUE |
server status: 0(4) | queue status: 0(Inf
) }
{ Resource: station3 | monitored: TRUE |
server status: 0(3) | queue status: 0(Inf
) }
```

4. PRODUCT TRAJECTORY AND ANALYZING STATIONS

Now that the system is in place, the product must flow in the system for manufacturing purpose, and the process path must be created this can be done with the help of trajectory function [4],[9].

```
> product <- trajectory(name = "Product P
ath", verbose = T)
> product
trajectory: Product Path, 0 activities
```

On station1 the raw material gets processed for 10 minutes, and on station2 the raw material gets processed for 15 minutes, and on station3 the raw material gets processed for 10 minutes [10].

```
> product %>%
+   seize("station1",1)%>%
+   timeout(function() rnorm(1,10))%>%
+   release("station1",1)%>%
+
+   seize("station2",1)%>%
+   timeout(function() rnorm(1,15))%>%
+   release("station2",1)%>%
+
+   seize("station3",1)%>%
+   timeout(function() rnorm(1,10))%>%
+   release("station3",1)

> time1 = Sys.time()
```

4.1 RUNNING SIMULATION

To begin the simulation, add_generator must be added so that the simulation will start, and the trajectory is given so that the raw material will flow through the system [7]. The interarrival time is given as 5 minutes with a standard deviation of 0.5 [7].

```
> sys%>%
+   add_generator(name_prefix = "sim_prod
uct",
+                 trajectory = product,
+                 distribution = function
()rnorm(1,5,0.5))
```

The time for the production system is eight hours, so the simulation of the system has to run for at least 480 minutes.

```
> sys %>% run(until = 480)

simmer environment: Production System | n
ow: 480 | next: 480.452233125804
{ Monitor: in memory }
{ Resource: station1 | monitored: TRUE |
server status: 2(3) | queue status: 0(Inf
) }
{ Resource: station2 | monitored: TRUE |
server status: 3(4) | queue status: 0(Inf
) }
{ Resource: station3 | monitored: TRUE |
server status: 2(3) | queue status: 0(Inf
) }
```

```
{ Source: sim_product | monitored: 1 | n_
generated: 96 }
```

One can see that in total 96 products are generated, and we c an also observe that station1 is occupied with 2 machines, st ation2 is occupied with 3 machines and station3 is occupied with 2 machines, and in all the 3 stations there is no queue. So, there is no bottleneck in these stations [11], [12].

5. ANALYZING RESOURCSE

Running simulation one time for one each resource will not benefit the system in any way so one has to do repeat the pro cess for all the resources at a time using functions in R progr amming, lapply function can be used to loop the system and in this case study it is advisable to run for at least 1000 simul ations [].

```
> product %>%
+   seize("station1",1)%>%
+   timeout(function() rnorm(1,10))%>%
+   release("station1",1)%>%
+
+   seize("station2",1)%>%
+   timeout(function() rnorm(1,15))%>%
+   release("station2",1)%>%
+
+   seize("station3",1)%>%
+   timeout(function() rnorm(1,10))%>%
+   release("station3",1)
```

Running the simulation repeatedly for 1000 cycles

```
> prosys <-lapply(1:1000, function(i){
+   simmer("Machine System")%>%
+     add_resource("station1",3)%>%
+     add_resource("station2",4)%>%
+     add_resource("station3",3)%>%
+     add_generator(name_prefix = "sim_pr
oduct",
+                   trajectory = product,
+                   distribution = functi
on()rnorm(1,5,0.5))%>%
+     run(480)%>%
+     wrap()
+ })

> time2 = Sys.time()
```

To see the time difference between simulation we can use thi s syntax

```
> time2-time1
Time difference of ------ mins
```

6. ANALYZING PLOTS

6.1 RESOURCE UTILIZATION AND USAGE OVER TIME

Now to find out the resource utilization, resource usage over time and the graph once can use the function get_mon_resou rces and to plot we use the plot function.

```
> resources <- get_mon_resources(prosys)
> p1= plot(resources, metric =  "utilizat
ion")
> p1
```

Below figure represents the utilization of resources i.e. three work stations and we can note from that graph that the maximum utilization is 75% for station2 and for other stations the utilization is 65% and this confirms that there is no bottleneck in the system and the flow of raw materials from stations is smooth [13], [14].

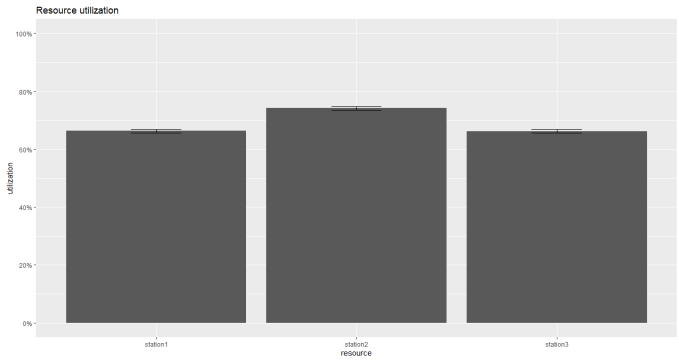

Figure 3: Resource Utilization

The second graph is the usage of resources over time, in this graph we can observe that station1 have 3 machines, station2 have 4 machines and station3 have 3 machines, the green li ne indicates the that in most cases the usage of machines has not exceeded overtime and all the stations are below usage a nd the red line indicates the queues where the queue never cr osses the resources and have no bottlenecks at all [13], [14].

```
> p2= plot(resources, metric= "usage",
+       c("station1","station2","station3"
),
+       items = c("queue","server"))
> p2
```

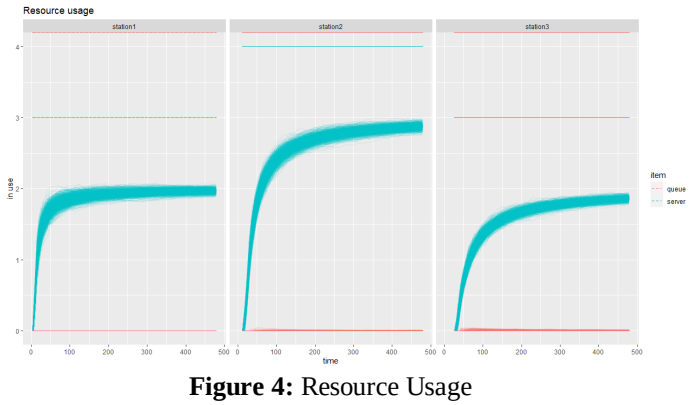

Figure 4: Resource Usage

6.2 RAW MATERIAL WAITING TIME AND ACTIVITY TIME

Now to find out the raw material waiting time, activity time and the graph once can use the function get_mon_arrivals an d to plot we use the plot function.

```
> arrivals <- get_mon_arrivals(prosys)
> p3= plot(arrivals, metric =  "waiting_t
ime")
> p3
```

Below is the graph for the waiting time of raw materials and it indicates that there is no problem for the raw material to enter into the system since there is no increase of waiting time with the increase of flow time/ simulation time as system move forward [10], [12].

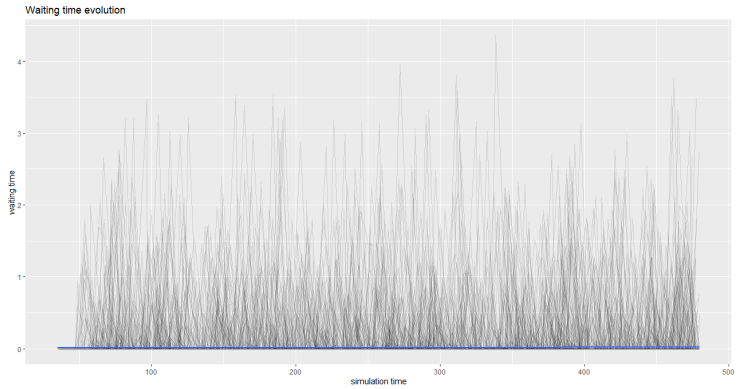

Figure 5: Waiting time for Raw Materials

From activity time one can observe that activity of raw material is constant over time of and there is no problem with the raw material.

```
> p4= plot(arrivals, metric =  "activity_
time")
> p4
```

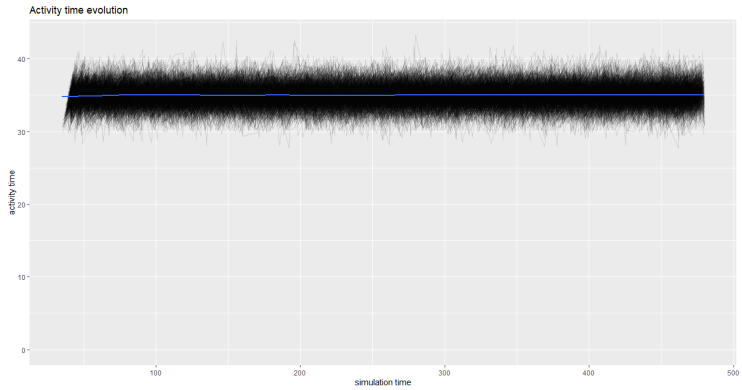

Figure 6: Activity Time

```
> library(gridExtra)
> grid.arrange(p1,p2,p3,p4)
```

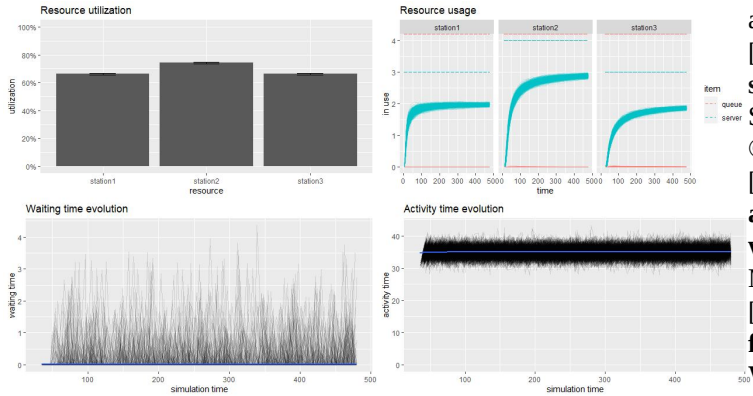

Figure 7: Complete Grid of Plots

Above graph represents the complete grid of graphs for overall together for this a library must be installed "gridExtra", so that all the plots appear in a single source format.

7. CONCLUSION

R Programming language comes very handy and is development of syntaxes for Discrete event simulation is becoming quite popular. The goal is to custom R programming language for DES with the help of industrial production data for post-processing analysis. There are lot of features that R programming can be implemented for DES.

8. FUTURE SCOPE

There is a lot of potential to use R Programming to use in Discrete Event Simulations, the complexity of real-world systems with unaffordable analytical models make it difficult for the simulation studies, there are several studies that indicate the Discrete event simulations and scheduling where complex statistical calculations are to be done in order to solve these time consuming problem.

REFERENCES

[1] Phanindra Kshatra D., Ratna Prasad P., Akhil S., Charan Raghavendra B.L.N.S.S., Prakash S.V.S. (2019), **'A methodology to adapt and understand a manufacturing system and operations using discrete-event simulation'**, International Journal of Engineering and Advanced Technology, 8(6), PP.4998-5003.

[2] Phanindra Kshatra D., Ratna Prasad P., Kalamulla M., Sai Krishna P., Sai P.B.L.N. (2019), **'Analyze the production system of an body-in-white system through modelling and perform bottleneck, optimization using simulation software'**, International Journal of Recent Technology and Engineering, 8(2 Special issue 4), PP.685-690.

[3] Barry Lawson, Lawrence M. Leemis, **DISCRETE-EVENT SIMULATION USING R,** Proceedings of the 2015 Winter Simulation Conference, **978-1-4673-9743-8/15/$31.00 ©2015 IEEE**

[4] I~naki Ucar, Bart Smeets, Arturo Azcorra, **simmer: Discrete-Event Simulation for R**, arXiv:1705.09746 [stat.CO]

[5] Barry Lawson, Lawrence M. Leemis, **An R package for simulation education**, Proceedings of the 2017 Winter Simulation Conference, 978-1-5386-3428-8/17/$31.00 ©2017 IEEE.

[6] Sekar, M; Srinivas, J; Kotaiah, KR; Yang, SH , **Stability analysis of turning process with tailstock-supported workpiece**, International Journal of Advanced Manufacturing Technology, **43**, pages862–871(2009).

[7] Sarma, R; Rao, AVD; Girija, SVS, **On characteristic functions of the wrapped lognormal and the wrapped Weibull distributions**, Journal of Statistical Computation and Simulation, volume 81, 2011-Issue 5, Pages 579-589.

[8] Sastry, SS; Mallika, K; Rao, BGS; Tiong, HS; Lakshminarayana, S, **Identification of phase transition temperatures by statistical image analysis**, Liquid Crystals, Volume 39, 2012-Issue 6, Pages 695-700.

[9] Dama K.K., Surya S., Babu C., Pavan, Kiran R.(2018), **'Modal analysis of an automotive subsystem using cae techniques'**, International Journal of Mechanical Engineering and Technology ,9(3), PP. 1157-1162.

[10] Dr.M.Nageswara Rao, K.Dileep, Shaik Khadar Basha, Vara Kumari, **Modrak Algorithm to Minimize Completion Time for *n*-Jobs *m*-Machines Problem in Flexible Manufacturing System**, International Journal of Emerging Trends in Engineering Research, volume 8, no 8, August 2020.

[11] M. B. S. Sreekara Reddy, Ch. Dileep, A. Achyut Kumar, M. Raj, D. Sai Sandeep & Ch. Ratnam, **A Multi-Objective Hybrid Evolutionary Algorithm Approach to Address Flexible Job Shop Scheduling Problems**, International Journal of Mechanical and Production Engineering Research and Development (IJMPERD) ISSN (P): 2249-6890; ISSN (E): 2249-8001 Special Issue, Jun 2018, 199-208.

[12] Phanindra Kshatra D., Sai Krishna P., Sai P.B.L.N., Akhil Kumar B., Bhaskara Rao Y. (2019), '**Development of virtual modelling for optimization and botttleneck analysis of an automotive stamping using plant simulation'**, International Journal of Mechanical and Production Engineering Research and Development, 9(3), PP.931-938.

[13] M. B. S. Sreekara Reddy, Sourav Jena, V. Jayaram, P. Ravi, G. Advait & Ch. Ratnam, An Evolutionary **Algorithmic Approach To Solve Flexible Job Shop Problem Using Hybrid Genetic Algorithms**, International Journal Of Mechanical And Production Engineering Research And Development (Ijmperd) Issn (P): 2249-6890; Issn (E): 2249-8001 Special Issue, Jun 2018, 223-232.

[14] Pavan Kumar, **An inventory planning problem for time-varying linear demand and parabolic holding cost with salvage value,** Croatian Operational Research Review, CRORR 10(2019), 187-199.