World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:5, No:9, 2011

A New Heuristic Statistical Methodology for Optimizing Queuing Networks Using Discreet Event Simulation

Mohamad Mahdavi

***Abstract*—**Most of the real queuing systems include special properties and constraints, which can not be analyzed directly by using the results of solved classical queuing models. Lack of Markov chains features, unexponential patterns and service constraints, are the mentioned conditions. This paper represents an applied general algorithm for analysis and optimizing the queuing systems. The algorithm stages are described through a real case study. It is consisted of an almost completed non-Markov system with limited number of customers and capacities as well as lots of common exception of real queuing networks. Simulation is used for optimizing this system. So introduced stages over the following article include primary modeling, determining queuing system kinds, index defining, statistical analysis and goodness of fit test, validation of model and optimizing methods of system with simulation.

***Keywords*—**Estimation, queuing system, simulation model, probability distribution, non-Markov chain

I. INTRODUCTION

MOST of queuing models are solved by applying simplifying assumptions with classic mathematical methods, and optimized by systematic, heuristic or meta heuristic algorithms [3]. But most of these assumptions are not applicable and so is not possible to use mathematical analysis results. Real situation system optimizing is done based on cost function or productivity indexes, which is made more difficult by enlarged dimensions in real situation. For example M/M/1, a simple classic model, in which λ and μ are arrival rate on Poisson process and exponential service rate, respectively, is solvable and its productivity coefficient is $\rho = \frac{\lambda}{\mu}$. But in a model such as G/G/m/k/∞/Z the computations gets so complex and impossible to solve, in classic way [1].

In these cases, simulation tool can be very useful. But the manager of a system must be able to make the best decision for optimizing the queuing processes by using the most effective solution method. An applied algorithm for designing and optimizing queuing networks is presented in the first half of this paper.

On the other half (Section III to end), a real case study is done based on suggested algorithm step by step. It contains non-Markov features and the most special complex conditions like many other same cases and optimized by simulation.

Mohamad Mahdavi is Master of Industrial Engineering and Faculty Member of Industrial Engineering Department, Islamic Azad University, Najafabad Branch, Isfahan, Iran (corresponding author to provide cell phone: +98 913 331 4475; tell: +98 331 2291111; fax: +98 331 2648166; e-mail: mhdv412@yahoo.com).

II. THE ALGORITHM DEFINITION

The offered algorithm is shown in Figure 1. Based on the figure it starts from making single primary pattern of queuing system and ends with optimized design. After determining the type of arrival and service probability distributions in the whole system, it's possible to determine its approximate type. Then applicability of simplifying assumptions, considering real case conditions is evaluated. If it is not solved by classic models, simulation is used.

The first step of this mode is completing the primary pattern with whole details and defining suitable indexes for system evaluation which are based on cost or benefit only if their information is available, otherwise productivity indexes are usable. It's possible to change this model to run able program codes.

Model efficiency and credibility certainly must be got before applying optimizing strategies. It's possible to use different heuristic and meta heuristic algorithm as well as trial and error method and also sensitivity analysis for acquiring optimal solution.

In any steps of this algorithm different tools and techniques are usable and the best one is selected by analyzer according to the conditions.

III. THE CASE STUDY

This case is a network queuing system for transporting the minerals from a mine to the factory. In this process excavating is done by two same loader machines. The material is transported to the factory in six identical trucks. The trucks are marked from 1 to 6. After loading is ended, the trucks head toward scale station.

If the loaded weighs right it's transferred to the factory, otherwise, sent back to loading station to correcting. The trucks offload the material in the factory and go back to the mine and repeated frequently.

IV. SOLVING THE PROBLEM USING THE SUGGESTED ALGORITHM

A. *Providing a simple primary pattern*

There might be queuing in two parts of the process [1]. One is in loading station and another in scaling station.

However, in loading station the queues are divided to primary loading trucks and corrective loading ones. Their both capacity is infinite.

B. *Estimating* arrival and service probability functions

The following Probability Distribution Functions (PDF) determining is essential in thus process:

1. PDF of primary loading durations.
2. PDF of corrective loading durations.
3. PDF of scaling durations.
4. PDF of traveling durations to offloading.

As a sample we deal with PDF of primary loading:

a. Frequency Histogram

Figure 3 shows the frequency histogram of a 50-sample from primary loading duration in minutes. It's supposed to be similar to exponential distribution.

b. Probability Sheets

Relative probability sheet to this data for goodness of fit testing with Anderson Darling (AD) index and probability estimating in Maximum Likelihood Estimator (MLE) method is shown in Figure 4.

c. Coefficient of Variation (C.V)

To ensure the correction of estimated function, it is also better to use the C.V. Table I illustrates the C.V calculation results. The amount 0.894 which approaches 1 is suggested the exponential function for this data is desirable [2].

C. Determining kind of queuing system

Considering estimated PDFs and special conditions in the system, the queue kind in any of these stations is determined. For instance in the first station (just primary loading) the queue kind is almost like G/M/2/∞/6/Z, where G indicates the normality of arrival PDF and M indicates the exponential PDF for primary loading.
About arrival discipline (Z) the priority is with corrective trucks which sent back from scaling station 2 and 6 means there are 2 loader and 6 trucks [3].

D. Conforming with the classical models

When the queue kind in whole process is determined, the state of the system must be compared with classic models which are analyzed in theoretical arguments based on relative assumptions [3, 4]. If it is possible to make an exact conformity between them the classic results are used for solving the system. But as the results figure out, the existed queue system application doesn't include Markov features and isn't independent of its background and this chain doesn't make a Poisson process.

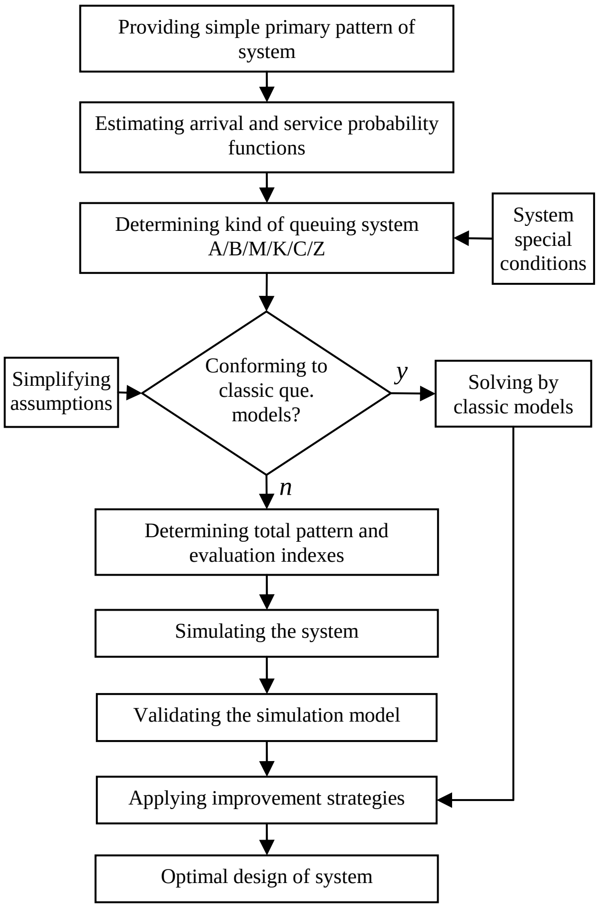


Fig. 1 Suggested Optimizing algorithm

On the other side due to limited capacity of scaling station as well as arrival priority discipline of primary loading, the Jackson network condition are not met so it is impossible to solve this system with classical Markov.

E. Total pattern and determining evaluation indexes

In this stage the primary simple pattern should be reviewed and completed with whole details. The appropriate indexes for evaluating and improvement measuring of the system should be defined according to analyzer. For instance in the under studying case, "the busy percent of loaders" can be defined. The simulation program should be able to compute the amount of evaluation indexes in each run [1].

F. Simulation the system

The system simulating is done in two general steps. Providing the manual diagrams for monitoring all the system events at the first, and making a computerized program which be able to run and replicate for a lot of times [8,10]. Fig 5 shows a sample result page of computer based code simulation bye Visual Basic.

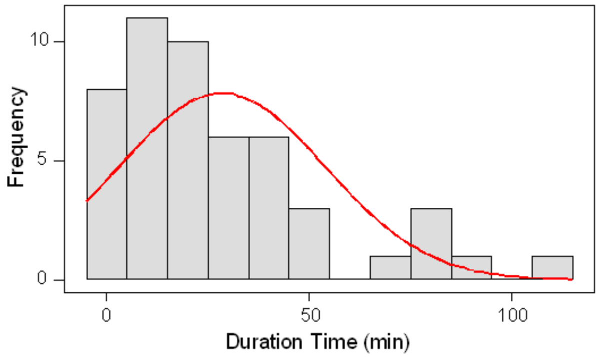


Fig. 2 Frequency histogram of primary loading duration times sample with normal curve

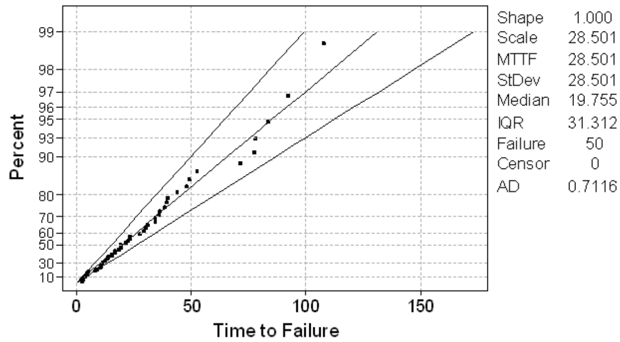


Fig. 3 Probability sheet for primary loading times to goodness of fit test

G. Validating the simulation model

The process of simulation model validation is so complex and important and there are many different methods for it. The Naylor and Fingure method is used in this case study.

For example, to comparing converting of simulation model's inputs to outputs with converting real system's inputs to output, a variable such as Y can be defined. So the waiting time per truck for primary loading in *i*th tripe is named by Y_i .

Then the equality examination between mean of model reply ($\overline{Y}$) and system reply (μ_y) is done by a test of hypothesis as bellow:

$$\begin{cases} H_0 : \mathrm{E(Y)} = \mu_y \\ H_1 : \mathrm{E(Y)} \neq \mu_y \end{cases}$$

Data of the real system is collected during an 8 hours work shift and the model results are obtained by 16 time replications of simulation model. If the H_0 is accepted, there is no significant reason for model invalidity [8].

The test results are presented in Table II and it is clear that based on this T sample hypothesis test, credibility of the model unable to be rejected.

H. Applying improvement strategies

For improving any process it should be measured. The main target of defining evaluation indexes in secession IV.E has also been the possibility of measurement.

According to amount of evaluation indexes (as the objective function), optimizing methods are able to applied. Even the low adequate method, trial and error, can be suitable sometimes. for example in here case study applying this method indicates that the best strategy is decreasing the primary loading duration time for 8 minutes and increasing 1 more scaling station [5].

I. Optimal design of system

The best and more reasonable method to improving system is using the cost-benefit functions. Because the goal of queuing system optimizing is finding their most capability, which has the minimum rate of customer waste time and maximum rate of service utilization.Although the productivity concepts can be useful, it's certainly obtained with a minimizing cost objective function. In such these minimizing problems the "servicing costs" and "customer idle time costs", which are related together indirectly, must be minimized [6, 7].

V. CONCLUSION

Because of many abnormal conditions in the real situation of queuing systems, most of assumptions for classical models are not provided and makes these models inapplicable. So the practical methods and algorithm are more useful. In this paper we represented an applied algorithm to optimizing complex queuing systems, that any kind of problems can be solved by the manager of a queuing network or system. So a real case study was used to defining the algorithm stages better and stepwise.

As we saw, this case had included most hard conditions to solve with classic models, especially non-Markov processes. Undoubtedly Most of the real queuing systems include special properties and constraints, which can not be analyzed directly by using the results of solved classical queuing models. Lack of Markov chains features, unexponential patterns and service constraints, are the mentioned conditions.

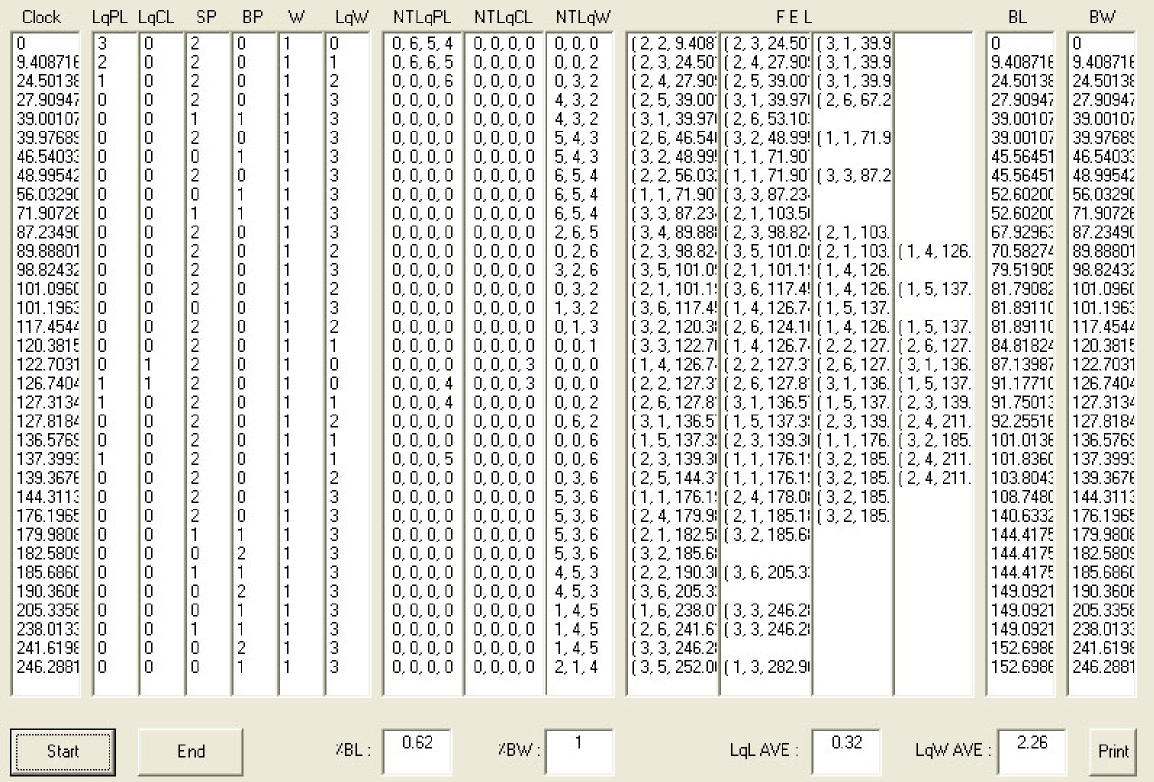
TABLE I
BASIC STATISTICS FOR PRIMARY LOADING DATA WITH C.V CALCULATION RESULT

Variable	*N*	*Mean*	*Median*	*StDev*	*Min*	*Max*	*C.V*
Primary Loading	50	28.50	20.35	25.48	2.36	108.15	0.894

TABLE II
RESULTS OF STATISTICAL ANALYSIS BASED ON T SAMPLE HYPOTHESIS TEST FOR VALIDATING OF SIMULATION MODEL

Test of mu = 49 vs mu not = 49

Variable	***N***	***Mean***	***StDev***	***SE Mean***	***95.0% CI***	***T***	***P-Value***	***H_0***
Y	16	47.06	10.47	2.62	(41.49 ; 52.64)	-0.74	0.470	Not Rejected



Clock	LqPL	LqCL	SP	BP	W	LqW	NTLqPL	NTLqCL	NTLqW	F E L				BL	BW
0	3	0	2	0	1	0	0, 6, 5, 4	0, 0, 0, 0	0, 0, 0	[2, 2, 9.408	[2, 3, 24.50	[3, 1, 39.9		0	0
9.408716	2	0	2	0	1	1	0, 6, 6, 5	0, 0, 0, 0	0, 0, 2	[2, 3, 24.50	[2, 4, 27.90	[3, 1, 39.9		9.408716	9.408716
24.50138	1	0	2	0	1	2	0, 0, 0, 6	0, 0, 0, 0	0, 3, 2	[2, 4, 27.90	[2, 5, 39.00	[3, 1, 39.9		24.50138	24.50138
27.90947	0	0	2	0	1	3	0, 0, 0, 0	0, 0, 0, 0	4, 3, 2	[2, 5, 39.00	[3, 1, 39.97	[2, 6, 67.2		27.90947	27.90947
39.00107	0	0	1	1	1	3	0, 0, 0, 0	0, 0, 0, 0	4, 3, 2	[3, 1, 39.97	[2, 6, 53.10			39.00107	39.00107
39.97689	0	0	2	0	1	3	0, 0, 0, 0	0, 0, 0, 0	5, 4, 3	[2, 6, 46.54	[3, 2, 48.99	[1, 1, 71.9		39.00107	39.97689
46.54033	0	0	0	1	1	3	0, 0, 0, 0	0, 0, 0, 0	5, 4, 3	[3, 2, 48.99	[1, 1, 71.90			45.56451	46.54033
48.99542	0	0	2	0	1	3	0, 0, 0, 0	0, 0, 0, 0	6, 5, 4	[2, 2, 56.03	[1, 1, 71.90	[3, 3, 87.2		45.56451	48.99542
56.03290	0	0	0	1	1	3	0, 0, 0, 0	0, 0, 0, 0	6, 5, 4	[1, 1, 71.90	[3, 3, 87.23			52.60200	56.03290
71.90726	0	0	1	1	1	3	0, 0, 0, 0	0, 0, 0, 0	6, 5, 4	[3, 3, 87.23	[2, 1, 103.5			52.60200	71.90726
87.23490	0	0	2	0	1	3	0, 0, 0, 0	0, 0, 0, 0	2, 6, 5	[3, 4, 89.88	[2, 3, 98.82	[2, 1, 103.		67.92963	87.23490
89.88801	0	0	2	0	1	2	0, 0, 0, 0	0, 0, 0, 0	0, 2, 6	[2, 3, 98.82	[3, 5, 101.0	[2, 1, 103.	[1, 4, 126.	70.58274	89.88801
98.82432	0	0	2	0	1	3	0, 0, 0, 0	0, 0, 0, 0	3, 2, 6	[3, 5, 101.0	[2, 1, 101.1	[1, 4, 126.		79.51905	98.82432
101.0960	0	0	2	0	1	2	0, 0, 0, 0	0, 0, 0, 0	0, 3, 2	[2, 1, 101.1	[3, 6, 117.4	[1, 4, 126.	[1, 5, 137.	81.79082	101.0960
101.1963	0	0	0	0	1	3	0, 0, 0, 0	0, 0, 0, 0	1, 3, 2	[3, 6, 117.4	[1, 4, 126.7	[1, 5, 137.		81.89110	101.1963
117.4544	0	0	2	0	1	2	0, 0, 0, 0	0, 0, 0, 0	0, 1, 3	[3, 2, 120.3	[2, 6, 124.1	[1, 4, 126.	[1, 5, 137.	81.89110	117.4544
120.3815	0	0	2	0	1	1	0, 0, 0, 0	0, 0, 0, 0	0, 0, 1	[3, 3, 122.7	[1, 4, 126.7	[2, 2, 127.	[2, 6, 127.	84.81824	120.3815
122.7031	0	1	2	0	1	0	0, 0, 0, 0	0, 0, 0, 3	0, 0, 0	[1, 4, 126.7	[2, 2, 127.3	[2, 6, 127.	[3, 1, 136.	87.13987	122.7031
126.7404	1	1	2	0	1	0	0, 0, 0, 4	0, 0, 0, 3	0, 0, 0	[2, 2, 127.3	[2, 6, 127.8	[3, 1, 136.	[1, 5, 137.	91.17710	126.7404
127.3134	1	0	2	0	1	1	0, 0, 0, 4	0, 0, 0, 0	0, 0, 2	[2, 6, 127.8	[3, 1, 136.5	[1, 5, 137.	[2, 3, 139.	91.75013	127.3134
127.8184	0	0	2	0	1	2	0, 0, 0, 0	0, 0, 0, 0	0, 6, 2	[3, 1, 136.5	[1, 5, 137.3	[2, 3, 139.	[2, 4, 211.	92.25516	127.8184
136.5769	0	0	2	0	1	1	0, 0, 0, 0	0, 0, 0, 0	0, 0, 6	[1, 5, 137.3	[2, 3, 139.3	[1, 1, 176.	[3, 2, 185.	101.0136	136.5769
137.3993	1	0	2	0	1	1	0, 0, 0, 5	0, 0, 0, 0	0, 0, 6	[2, 3, 139.3	[1, 1, 176.1	[3, 2, 185.	[2, 4, 211.	101.8360	137.3993
139.3676	0	0	2	0	1	2	0, 0, 0, 0	0, 0, 0, 0	0, 3, 6	[2, 5, 144.3	[1, 1, 176.1	[3, 2, 185.	[2, 4, 211.	103.8043	139.3676
144.3113	0	0	2	0	1	3	0, 0, 0, 0	0, 0, 0, 0	5, 3, 6	[1, 1, 176.1	[2, 4, 178.0	[3, 2, 185.		108.7480	144.3113
176.1965	0	0	2	0	1	3	0, 0, 0, 0	0, 0, 0, 0	5, 3, 6	[2, 4, 179.9	[2, 1, 185.1	[3, 2, 185.		140.6332	176.1965
179.9808	0	0	1	1	1	3	0, 0, 0, 0	0, 0, 0, 0	5, 3, 6	[2, 1, 182.5	[3, 2, 185.6			144.4175	179.9808
182.5809	0	0	0	2	1	3	0, 0, 0, 0	0, 0, 0, 0	5, 3, 6	[3, 2, 185.6				144.4175	182.5809
185.6860	0	0	1	1	1	3	0, 0, 0, 0	0, 0, 0, 0	4, 5, 3	[2, 2, 190.3	[3, 6, 205.3			144.4175	185.6860
190.3606	0	0	0	2	1	3	0, 0, 0, 0	0, 0, 0, 0	4, 5, 3	[3, 6, 205.3				149.0921	190.3606
205.3358	0	0	0	1	1	3	0, 0, 0, 0	0, 0, 0, 0	1, 4, 5	[1, 6, 238.0	[3, 3, 246.2			149.0921	205.3358
238.0133	0	0	1	1	1	3	0, 0, 0, 0	0, 0, 0, 0	1, 4, 5	[2, 6, 241.6	[3, 3, 246.2			149.0921	238.0133
241.6198	0	0	0	2	1	3	0, 0, 0, 0	0, 0, 0, 0	1, 4, 5	[3, 3, 246.2				152.6986	241.6198
246.2881	0	0	0	1	1	3	0, 0, 0, 0	0, 0, 0, 0	2, 1, 4	[3, 5, 252.0	[1, 3, 282.9			152.6986	246.2881

Start End %BL : 0.62 %BW : 1 LqL AVE : 0.32 LqW AVE : 2.26 Print

Fig. 4 A sample of result sheet for simulation program

REFERENCES

[1] Jerry Banks, John S. Carson; Discrete-Event System Simulation, Prentice-Hall INC., 1984.

[2] Albert H. Bowker, Gerald J. Lieberman; Engineering Statistics, 2nd Edition, Prentice-Hall INC., 1972.

[3] Gross, D., C. M. Harris; Fundamentals of Queuing Systems, 2nd edition, John Wiley, New York, 1984.

[4] Kleinrock, L., Queuing systems, (Vol. 1: Theory), John Wiley, New York, 1975.

[5] Law, A. M., W. D. Kelton; Simulation Modeling and Analysis, McGraw-Hill, New York, 1982

[6] Box, G. E. P., M. F. Muller; “A Note on the Generation of Random Normal Deviates”, Annals of Mathematical Statistics, 610-11, Vol.29, 1958.

[7] Marse, K., Roberts; “Implementing a Portable FORTRUN Uniform (0, 1) Generator”, Simulation,Vol. 41, 135-139, 1983

[8] Naylor, T.H., J. M.Finger, “Verification of Computer Simulation Model”, Management Science, Vol. 14, PP.92 -101, 1967.

[9] Schruben, L. W., “Establishing the Credibility of Simulations”, Simulation, Vol. 34, PP. 101-5, 1980.

[10] Son, Y., Wysk, R. A., Jones, A. T.; “Simulation-based shop floor control: Formal model, generation and control interface”, IIE Transactions on Design and Manufacturing, Vol. 35, PP. 29-48, 2003.