

# Modeling and Simulation of Queuing Systems Using Stochastic Petri net and Arena Software: A Case Study

Husam Kaid<sup>1,a</sup>, Abdulmajeed Dabwan<sup>1,b</sup>, and Abdulrahman Al-Ahmari<sup>1,2,c</sup>

1. College of Engineering, Industrial Engineering Department, King Saud University, Riyadh 11421, Saudi Arabia
2. FARCAMT Chair, Advanced Manufacturing Institute, King Saud University, Riyadh 11421, Saudi Arabia

[yemenhussam@yahoo.com](mailto:yemenhussam@yahoo.com), [abmj999@yahoo.com](mailto:abmj999@yahoo.com), [alahmari@ksu.edu.sa](mailto:alahmari@ksu.edu.sa)

## Abstract

Stochastic Petri nets (SPN) and Arena simulation software can both be used to study the qualitative and quantitative behavior of systems in a single environment. However, no comparative studies of the two formalisms have been conducted. Therefore, this paper proposes a methodology to evaluate the selected formalisms by using the ViseKriterijumska Optimizacija I Kompromisno Resenje (VIKOR) method, which can provide a better understanding of both formalisms and prepare a fertile ground for exchanging ideas and techniques between the two. A real world system of the main student restaurant of the King Saud University in Saudi Arabia is discussed in this paper to demonstrate the applicability of the proposed methodology. First, data for the study is collected, and thereafter, the appropriate distribution is adapted for the data points using the Arena input analyzer. A visual network was then developed and run on the stochastic Petri nets and Arena simulation software in order to acquire output for the simulation. The results showed that the Arena and SPN simulation software are convenient for modeling and analyzing the case study. However, using the VIKOR approach, a comparative analysis showed that SPN is better than Arena.

**Keywords:** Queuing systems, Arena, Stochastic Petri net, VIKOR

## 1. Introduction

Simulation is the imitation of the operation of a real-world process or system over time (Banks et al. 2005). A descriptive technique of simulation enables a decision-maker to evaluate the behavior of a model under various conditions (Stevenson 2005). However, before conducting any simulation, a model must be developed. This model represents the key characteristics or behaviors of a selected physical or abstract system (Gallagher and O'Sullivan 2011). The simulation represents the operation of the system over time, whereas the model represents the system itself (Levine et al. 2013). Simulation is used in many contexts, such as safety engineering, training, simulation of technology for performance optimization, education, testing, and even in video games (Banks et al. 2005). Training simulators, which include flight simulators, are utilized for training aircraft pilots and provide them with a virtual experience (Kelton et al.). Simulation can be used to show the eventual effects of a course of action and alternative conditions (Law 2007). It can also be used with human systems to gain insight into their functions, or for scientific modeling of natural systems. Simulation is also used when the real system cannot be engaged because it is inaccessible, unacceptable, dangerous, and inexistent, or it may be in the process of being designed, but not yet built (Law 2007). A computer simulation is an attempt to model and study hypothetical or real-life situations on a computer in order to visualize how the system works (Banks et al. 2005). Predictions about the behavior of a system can be made by changing the variables in the simulation. It is a tool for the virtual investigation of the behavior of a system under study. Simulation can be defined as a series of "what-if" type of experiments that are implemented on the simulation model. Evidently, simulation has a wide range of applications (Law 2007). In relation to the foregoing, there are several commercial off-the-shelf simulation software that can be used to develop a simulation model. Two of these are the stochastic Petri nets (SPN) and Arena software.

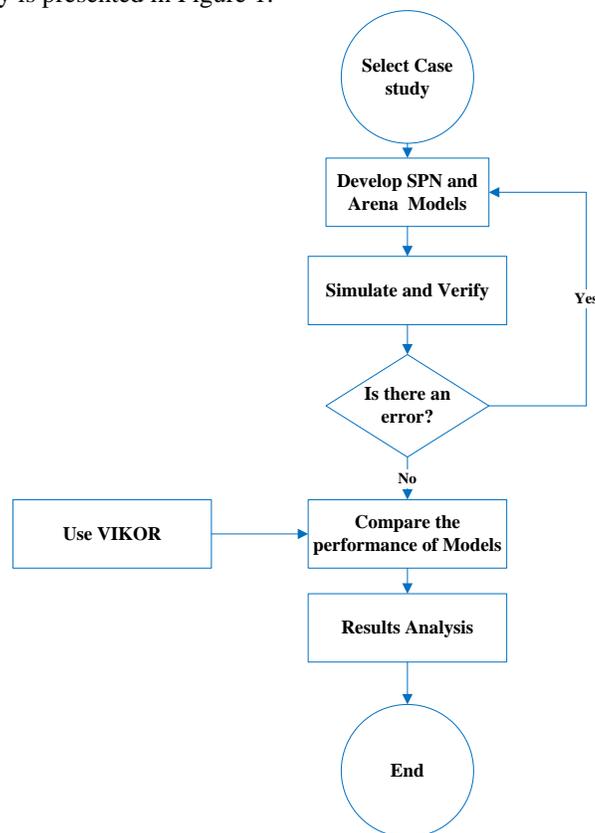
Stochastic Petri nets is one of the several mathematical modeling languages that can be used for describing distributed systems. Petri nets (PN) is a valuable formalism to model manufacturing systems and have been used in a plethora of

industrial problems, such as in the construction of schedules (Sawhney 1997), scheduling and control of semiconductor manufacturing systems (Zhou 1998), model verification and validation (Samkari and Franz 2012), and manufacturing systems and deadlock prevention (Li and Zhou 2004, Abdulaziz et al. 2015, Nasr et al. 2015). On the other hand, the Arena software enables users to bring the power of modeling and simulation into their business. It is designed for analyzing the impact of changes involving significant and complex redesigns associated with the supply chain, manufacturing, processes, logistics, distribution and warehousing, and service systems. The software provides maximum flexibility and range of application coverage for modeling at any desired level of detail and complexity (Altiok and Melamed 2010). However, comparative studies between the Arena and stochastic Petri net software were never conducted. Therefore, in this paper a methodology is proposed to evaluate the selected formalisms by using ViseKriterijumska Optimizacija I Kompromisno Resenje (VIKOR) method. The proposed method can provide a better understanding of the two formalisms, as well as prepare a fertile ground for the exchange of ideas and techniques between the two.

Commencing with this introductory section, the paper is organized as follows. Section 2 describes the proposed methodology. Section 3 displays the results of using the proposed methodology on a real case study. Section 4 presents results and discussions, and Section 5 introduces the comparative analysis of SPN and Arena. Finally, the conclusion and future work are presented in Section 6.

## 2. Methodology

This section discusses the methodology proposed in this study. The first section illustrates the Petri net concepts, and the second section describes the Arena concepts. The last section presents the VIKOR approach. The flowchart of the implementation methodology is presented in Figure 1.

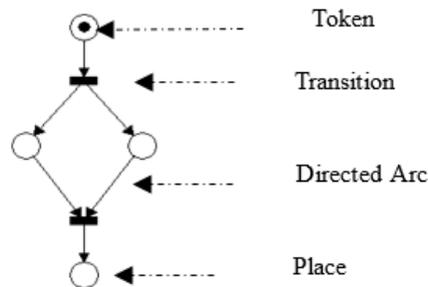


**Figure 1. Flowchart of the proposed methodology**

### 2.1 Basics of Petri net

The Petri net theory was originally developed by Carl Adam Petri in 1962, and was presented as part of his dissertation. The common definition of PNs is introduced by Petri. A Petri net or place/transition net can be defined as a five-tuple in the form  $PN = (P, T, I, O, Mo)$  (Nasr et al. 2015), where P and T are finite non-empty sets of places pictured by

circles and transitions pictured by bars, respectively. The input function,  $I: P \times T \rightarrow \{0, 1\}$ , defines the set of directed arcs from  $P$  to  $T$ . On the other hand, the output function,  $O: P \times T \rightarrow \{0, 1\}$ , defines the set of directed arcs from  $T$  to  $P$ . Moreover,  $M_0: P \rightarrow N$  is a marking whose  $i^{\text{th}}$  component represents the number of tokens in the  $i^{\text{th}}$  place,  $P_i$ . The disjoint sets are  $P$  and  $T$ , and  $P \cup T$  are called nodes with  $P \cup T \neq \emptyset$  and  $P \cap T = \emptyset$ . Petri nets are assumed to be connected, which means that there is at least one path between any two nodes. Evidently, the input and output functions of Petri nets can be represented by arcs with arrows between two different types of nodes. An inhibitor arc,  $In$ , is represented by an arc with a small circle (not an arrow). An inhibitor arc connects an input place  $p$  to a transition  $t$ . Transition  $t$  is enabled if the input place  $p$  has tokens less than the inhibitor arc weight  $In(p, t)$ . On the other hand,  $En$  is an enabling arc represented by an arc with an empty arrow. An enabling arc connects an input place  $p$  to a transition  $t$ . Transition  $t$  is enabled if the input place  $p$  has tokens that are at least equal to the enabling arc weight  $En(p, t)$ . The initial marking of the net is denoted by  $M_0$ , which represents the different raw parts that are to be synchronously processed in the system, as well as the state of resources, such as machines and robots (Abdulaziz et al. 2015). A marked PN and its elements are shown in Figure 2.



**Figure 2: A simple Petri net**

The SPNs and generalized SPNs (GSPNs) are two popular extensions of Petri nets (Kaid et al. 2015). Both have been considerably used in modeling, controllers, and analysis of manufacturing systems. There are several studies in the areas of the manufacturing system. Zhou et al. (1990) used stochastic PN modeling based on the top-down and bottom-up approaches to analyze and compare the performance of a flexible manufacturing system (FMS) under both deadlock-free and deadlock-prone systems. The study of Al-Jaar and Desrochers (1990) uses GSPN modules as basic building blocks to model and analyze complex manufacturing systems. A new extended stochastic high-level evaluation Petri nets presented by Yan et al. (1998) are more suitable for the modeling and simulation of FMSs. Generic stochastic colored Petri net submodels of machines and conveyor systems were developed by Moore and Gupta (1995). Zimmermann et al. (2001) presented the GSPN approach for modeling and analyzing tandem automated guided vehicle systems. Liu et al. (2008) examined the modeling and performance analysis of production lines, consisting of two machines subject to failures. Patel and Joshi (2013) developed a GSPN model for a manufacturing system with a deadlock to generate the reachability tree. Coman and Ionescu (2014) used SPNs to evaluate performance measures, such as the utilization rates of machines, deadlock detection, cycle time, and throughput rate of a system. Similarly, Tigane et al. (2017) developed reconfigurable SPNs for reconfigurable manufacturing systems.

## 2.2 Arena

Arena® is a worldwide leader among discrete event simulation (DES) software. It is a graphical interface, which uses the SIMAN language. SIMAN is a low-level simulation language, which provides the basic elements for DES, such as events list management, generation of random numbers, and collection of data for statistical purposes. The simplest modules in Arena are elements and blocks (Ahtiok and Melamed 2010). The elements represent basic language structures. The main Arena elements are as follows:

- Entities: These are structures that flow across modules. Each entity moves when an event is triggered.
- Resources: These are elements required to perform a task.
- Queues: These are elements in which the entities wait for the availability of resources.
- Variables: These are global data structures, which can be accessed and modified by the entities.
- Attributes: These elements are local variables specific to individual entities. An attribute might be the skin color or sex. A variable might be the number of male pedestrians, the number of female pedestrians, etc.

Blocks are intended to perform actions on the elements. Typical actions change the values of entities and/or variables, including making decisions, such as picking a specific queue and controlling the flow of entities. The blocks used in this application are the following:

- Create: Creates new entities. In a PN, it is used to create the initial marking.
- Branch/Decide: Picks a path depending on a condition, e.g., priority or probability.
- Hold and Signal. The hold block stops the flow of an entity until a condition is satisfied. When such a condition is satisfied, the signal block issues a message to the hold block and the entity is released. This block can be used to move the tokens from one place to a transition.
- Assign: This block changes the values of variables and/or attributes.
- Separate: This block is used to duplicate entities.

Figure 3 shows some of the blocks of Arena.

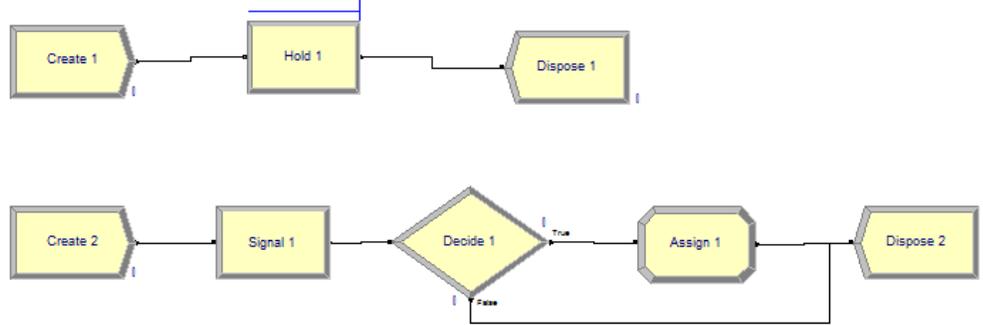


Figure 3. Some blocks of Arena

### 2.3 VIKOR Approach

Multiple-criteria decision-making (MCDM) approaches are methods particularly used for solving conflicts between different management and engineering topics (Deng and Chan 2011). The VIKOR method is a compromise ranking method for optimizing the decision-making process. VIKOR concentrates on ranking and selecting from a set of alternatives in the presence of criteria (Kang and Park 2014). Thus, the VIKOR technique is found to be an appropriate decision method (Akman 2015). The technique has the following steps:

*Step 1:* Assign values for the linguistic variables in relation to sustainable process alternatives. Accordingly, a matrix of alternatives has been developed for each criterion.

*Step 2:* Form the decision matrix. The ratings of maintainable processes (alternatives) resulting from ratings of experts are collected. Thereafter, the decision matrix is constructed using Eq. (16).

$$A = \frac{1}{k} \sum_{k=1}^k A_k \quad (1)$$

*Step 3:* Identify the best  $f_b^*$  and the worst  $f_b^-$  values of all criterion ratings,  $b = 1, 2, \dots, n$ . These values are computed using Eqs. (17) and (18):

$$f_b^* = \max(f_{ab}) \quad (2)$$

$$f_b^- = \min(f_{ab}) \quad (3)$$

where  $f_b^*$  and  $f_b^-$  are the positive and negative ideal solutions for the  $b$ th criterion, respectively.

*Step 4:* Compute the values of  $S_a$  and  $R_a$  for  $a = 1, 2, \dots, m$  by Eqs. (19) and (20):

$$S_a = \sum_{b=1}^n W_b \left[ \frac{f_b^* - f_{ab}}{f_b^* - f_b^-} \right] \quad (4)$$

$$R_a = \max_b \left[ W_b \left[ \frac{f_b^* - f_{ab}}{f_b^* - f_b^-} \right] \right] \quad (5)$$

where  $S_a$  refers to the rate of the distance to the positive ideal solution, i.e., the maximum “group utility of majority”;  $R_a$  is the rate of the distance to the negative ideal solution, i.e., minimum “individual regret of opponent” in the compromise programming technique to aid in defining compromise solutions based on parleyed predilections of decision-makers;  $W_b$  refers to the weight every criterion.

Step 5: Define the aggregating index, i.e., final negotiation solution,  $Q_a$ , for  $a=1, 2, \dots, m$  by Eq. (21). The alternative, which has a minimum  $Q_a$ , is the best alternative. This negotiation solution must be steady in a decision-making process. This negotiation solution is given by the following:

$$Q_a = v \frac{S_a - S^*}{S^- - S^*} + (1 - v) \frac{R_a - R^*}{R^- - R^*} \quad (6)$$

where

$$S^- = \max_a S_a S^- \quad (7)$$

$$S^* = \min_a S_a S^* \quad (8)$$

$$R^- = \max_a R_a R^- \quad (9)$$

$$R^* = \min_a R_a R^* \quad (10)$$

Moreover, the solutions derived from  $\max_a S_a$  show the “maximum group majority” and the solutions derived from  $\min_a R_a$  show the “minimum individual regret” of the alternative. The weight of the path of action or maximum set usefulness is denoted by  $v$ , and  $(1 - v)$  point to the weight of the individual remorse. The value of  $v$  in this study is 0.5.

Step 6: Rank the alternatives based on  $Q_a$  values.

Step 7: Define the minimum value of  $Q$ . The minimum value of  $Q$  is recommended as the best negotiation solution if it satisfies the following two conditions:

**Condition 1:** If  $Q(A^{(2)}) - Q(A^{(1)}) \geq 1/n - 1$ , then the alternative  $Q(A^{(1)})$  shows a suitable advantage, where  $A^{(1)}$  and  $A^{(2)}$  are the alternatives and  $n$  is the number of alternatives.

**Condition 2:** The alternative  $Q(A^{(1)})$  is steady in decision-making when it is also best ranked in  $S_a$  and  $R_a$ .

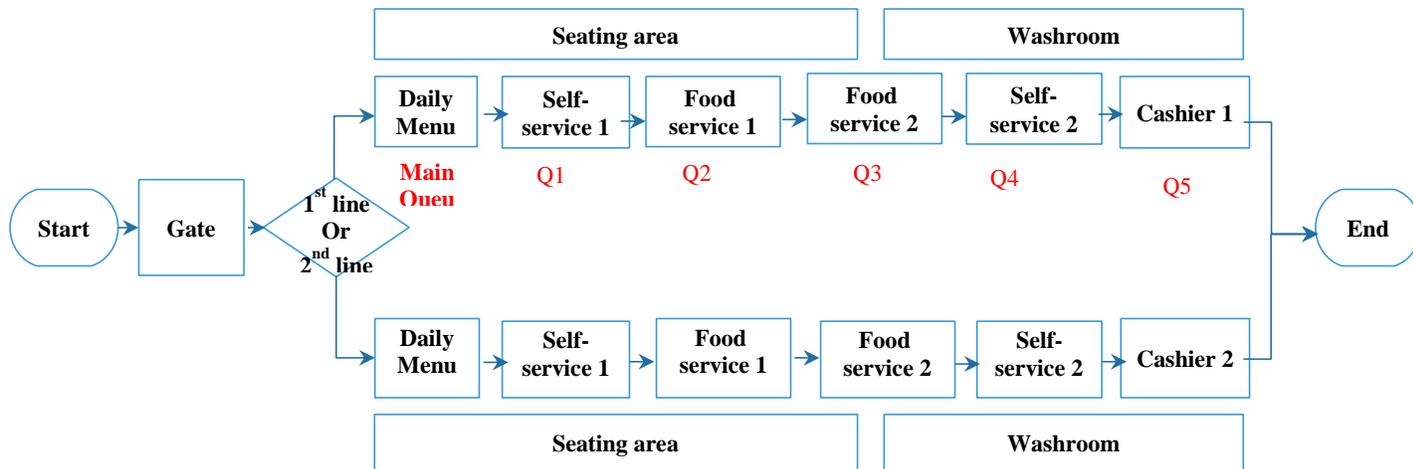
Step 8: Determine the best alternative by choosing  $(A^{(m)})$  as the best negotiation solution with the least value of  $Q_a$  with regard to the conditions mentioned above, where  $m$  represents the selection alternatives, i.e.,  $S_1, S_2, \dots, S_m$ .

### 3. Case Study

The main problem in the King Saud University (**KSU**) restaurant system is the daily long waiting time, except on Thursdays, specifically, during lunch time. Because of the long waiting time, there are instances when the queue can reach even as far as outside the main gates of the restaurant during rush hours. Accordingly, the objective of this study is to model, analyze, and improve the system during rush hours using the **Petri nets** and **Arena** simulation software. In order to reduce the waiting time, part of the objective is to scheme a set of alternatives that may improve the efficiency of the system and to obtain better service quality during rush hours. The main KSU restaurant has two identical serving lines, each with a single queue. The restaurant layout is shown in Figure 4. Each line consists of six sections. The first section is where the daily menu, food utensils and dishes are located. It is here where students choose from the menu before proceeding to the next section. The second section is a self-service area, where a student selects from a small menu of appetizers and snacks, such as salads, desserts, fruits, yogurt, and Laban. Afterwards, the student moves to the food-service area. In this section, two food attendants serve a menu of rice, meat or fish, chicken, soups or whatever the menu consists of each day. The menu is changed daily in order to deliver a variety of foods for the students. The first food attendant serves rice only, whereas the second serves the rest of the menu. Thereafter, the student proceeds to the fourth section, which is another self-service area, where student can get bread and condiments, such as ketchup, salt, and pepper. In the last section (cashier area), the cashier asks for the student's ID and counts the number of items selected by the student, who is then charged 4 SAR for a regular meal.

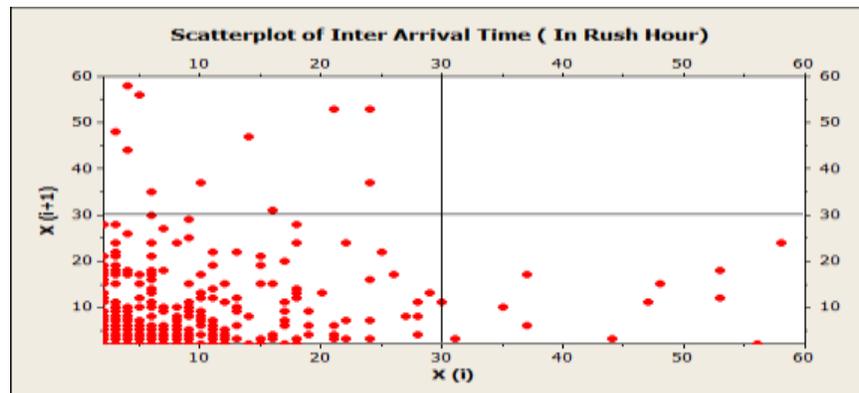
#### 3.1 Model Building and Data Gathering

As previously mentioned, the main KSU restaurant follows a queuing system with two identical service lines. Each line consists of five sections. Students who arrive at the restaurant can select either one of the two lines (Area 1 line 1 and Area 2 line 2). Consequently, if all lines are busy, the students will have to wait in a queue in front of any of the two lines. The main restaurant is characterized by three components: the arrival process, service mechanism, and queue discipline. The arrival process is a description of the arrival of students into the system. The service mechanism is expressed by specifying the number of servers and the probability distribution of the service times of students. The queue discipline refers to the first-in-first-out rule, and that each line has its own queue. The distribution of the service times ( $S_i$ ) needs to be specified for each part of the service in each line (self-service parts 1 and 2 and two food attendants in the food service parts of each line).



**Figure 4. Layout and lines of the KSU restaurant**

In addition, the distribution of interarrival time needs to be specified for the system. In addition, that what will be done in the data gathering and test? Data were collected over a period of 5 days. Data collected on interarrival times (in seconds) for one periods from 12:00 to 1:00 pm, which is the rush hour period. Proposed inputs data are as follow: Time between arrivals for the students, percentage of the students chooses first or second area, Time of service in the first self-service part in each line, Service time for each one of the two servers in food-service part in each line, Time of service in third part of the lines (second self-service), and Service time for cahier in each line. The data collected tested and summarized in a meaningful way that would allow us to determine the associated distributions of arrival and service times which is the inputs of our simulation model. The data were tested using scatter diagram, correlation plot, and Chi Square and Kolmogorov Smirnov tests. Minitab and Arena input analyzer softwares used to do so. Some of the tests is a scatterplot and statistic of inter-arrival time (Minitab Software) as shown in Figure 5 and Table 1, respectively. The rest estimation of parameters and goodness of fit tests are summarized in Table 2.



**Figure 5: Scatterplot of inter-arrival time (Minitab Software)**

**Table 1. Statistics of interarrival time (Arena input analyzer)**

Statistic	Value
<b>Distribution:</b>	Weibull
<b>Expression:</b>	1.5 + WEIB(5.45, 0.826)
<b>Square Error:</b>	0
<b>Chi Square Test</b>	
<b>Number of intervals</b>	17
<b>Degrees of freedom</b>	14
<b>Test Statistic</b>	22.8 < 23.685 at 95% confidence level

<b>Corresponding p-value</b>	= 0.169 > $\alpha = 0.05$
<b>Data Summary</b>	
<b>Number of Data Points</b>	478
<b>Min Data Value</b>	2
<b>Max Data Value</b>	58
<b>Sample Mean</b>	7.62
<b>Sample Std Dev</b>	8.34

**Table 2. Estimation of parameters and goodness of fit tests**

<b>Proposed inputs data</b>	<b>Distribution</b>	<b>Expression</b>
Time between Arrivals	Weibull	1.5 + WEIB(5.45, 0.826)
Cashier of Line1	Lognormal	3.5 + LOGN(12.2, 12.6)
Cashier of Line2	Lognormal	4.5 + LOGN(12.3, 11.7)
Line1 Food Server1	Gamma	4.5 + GAMM(5.89, 1.99)
Line1 Food Server2	Erlang	5.5 + ERLA(7.96, 2)
Line1 Self-service 1	Triangular	TRIA(17.3, 18.6, 20)
Line1 Self-service 2	Triangular	TRIA(12, 12.5, 18)
Line2 Food Server1	Lognormal	6.5 + LOGN(10.4, 8.22)
Line2 Food Server2	Gamma	3.5 + GAMM(6.91, 2.57)
Line2 Self-service 1	Triangular	TRIA(13.2, 14.8, 16.8)
Line2 Self-service 2	Triangular	TRIA(12, 12.5, 18)

### 3.2 Simulation Models Building and Description

The Arena and SPN models are shown in Figure 6 and 7, respectively. The entities that were modeled in this system are the students. The analysis of the system is conducted during the lunch period, which is from 12:00–1:00 pm. A CREATE module is developed to model the arrival of students. Because students who arrive into the system can choose to go either to the first or second line, the DECIDE module is used to test the conditions successively. After choosing a line the students join its queue. If there are any students before them in the queue, they need to wait for their turn to be served. If there is no one in the queue, they will be served immediately (First in First Out Queue). Each queue in front of each line is represented by the HOLD module, which is a queue with a blocking condition. After joining and waiting in the queue, the students will be served. Because the system has two identical lines, the general processes in each line are explained based only on one line, i.e., line 1. The first module in the line is the PROCESS module, which represents the first part of the self-service process. We used the options for seizing the resource that can delay the entities within a specified time in order to consider the appropriate statistical expressions, and make the resource busy until the entity passes to another PROCESS model, at which time the resource is released. After completing the first self-service, the student moves to the first food server, if the server is not occupied. To represent the buffer area between the two servers with a limited number of students in the queues, we also used the PROCESS module (if the food server 1\_1 is busy, the student is blocked between the two processes). After passing through all five services (self-service 1, food service 1, food service 2, self-service 2, and cashier service), the student leaves the system, and locates a seat to take lunch. This action can be represented by a DISPOSE module.

### 3.3 Verification and Validation

Pilot runs were used to validate the proposed models. This process need to be performed to ensure that the simulation model developed is valid and acceptable before proceeding to the succeeding steps. Verification seeks to show that the computer program performs as expected and intended. Validation on the other hand, seeks to determine whether the model behavior validly represents that of the real world system being simulated. Consequently, the model was run 90 times (90 replications). The results were used to verify and validate that the model was built as follows: (1) The process steps are similar between the model and real system; (2) The compiler in the program was used to check that the program run is correct; (3) It is known that there is no error during

program execution; (4) The outputs between the proposed models and real system were compared. A commonly used validation tolerance is 10%, which means that the output obtained from the simulation model must not exceed 10% of the real system output. This process is quite difficult to perform but needs to be executed in order to obtain a successful model. Table 3 summarizes the comparisons between the simulated and real system outputs for the average waiting time, which are very close to each other and with a maximum difference of 4.68%. Accordingly, the developed simulation models are considered as valid and acceptable. The next step implied that the proposed improvement model for the simulation model be developed.

**Table 3. Verification and Validation Result**

<b>Model type</b>	<b>Experiment</b>	<b>Sample Mean (Real System)</b>	<b>Sample Mean (Simulation Model)</b>	<b>Deviation</b>
<b>Arena model</b>	<b>Line 1</b>	13.55	14.1843	4.68%
	<b>Line 2</b>	13.90	14.1282	1.64%
<b>SPN model</b>	<b>Line 1</b>	13.55	13.19	-2.65%
	<b>Line 2</b>	13.90	13.81	-0.6%

#### 4. Results and Discussion

Waiting time is the time required for a student to wait for the services needed. Table 4 lists the results of the simulation model obtained from the Arena and SPN reports. Both indicate the existence of a long waiting time in line 1, with averages of 14.1282 and 13.19 min per student for Arena and SPN models, respectively. For line 2, the waiting time has averages of 14.1282 and 13.81 min per student for Arena and SPN models, respectively. Evidently, a long waiting time is a frequent and common complaint among students than complaints on other phases. The proposed model is conducted using a “what-if analysis.” The model experiment involves a test or a series of tests, considering a number of changes made to the input variables for the simulation model. Here, one suggested experimentation scenario is performed into the models to verify whether the proposed experimentation is capable of reducing the average waiting time or not. In the scenario, change is made in the self-service process by removing the second self-service. This involves moving cold drinks to the first self-service section, and food utensils and condiments after the cashier section. The proposed experimentation model is executed with 90 replications. The average waiting time under those replications is listed in Table 4. Based on the results on this table, the new average waiting time in line 1 is 13.82 and 12.00145 min per student for Arena and SPN models, respectively. For line 2, the waiting time has averages of 13.65 and 12.8154 min per student for Arena and SPN models, respectively. Compared with the previous (original) model (simulation output) results, the system is improved. This shows that the average waiting time per student can be reduced if the second self-service is removed and amenities in it are moved to a location after the cashier section. From the above results, it can be concluded that the proposed experimentation models exhibited a better reduction in the waiting time of students. Figure 8 and Figure 9 show the time in the system for lines 1 and 2, respectively. A comparative analysis of the proposed models was conducted to evaluate the suggested techniques. In the comparison between the two techniques with respect to structural and computational complexity, SPN have emphasized the flexibility to express important synchronization events directly into the model, at the expense of solution efficiency. This technique can be more appealing when studying new problem domains in which synchronization (actual or potential) performs a significant role in determining system performance, and where the potential inaccuracy of the approximate representation of these effects in the Arena model is unacceptable. In addition, in the comparative analysis with respect to the agility in adding new resources or activities, SPN similarly have emphasized the flexibility to respond to any modifications in the system.

**Table 4. Arena and SPN results**

Measure	Current system				Improved system			
	Arena Model		SPN Model		Arena Model		SPN Model	
	Line 1	Line 2	Line 1	Line 2	Line 1	Line 2	Line 1	Line 2
<b>Time in the system (min)</b>	14.1843	14.1282	13.19	13.81	13.82	13.65	12.00145	12.8154
<b>Throughput (student)</b>	138	144	127	112	141	131	123	122
<b>Utilization Server 1 (%)</b>	96.16	81.78	82.85	76.45	96.25	82.16	96.86	95.93
<b>Utilization Server 2 (%)</b>	98.15	81.87	84.79	73.87	97.97	82.52	97.77	86.411
<b>Utilization Cashier (%)</b>	57.64	62.06	72.55	68.52	58.17	63.07	74.544	72.55

### 5. Comparative analysis of SPN and Arena

In this section, the stochastic Petri nets and Arena are compared from the point of view of the model construction. A more informal comparison is conducted with the aim of providing insight into the relative strengths and weaknesses of the two paradigms. A comparative analysis was also conducted using VIKOR. The criteria used to evaluate the best simulation techniques are divided into running times, accuracy, construction of scenarios, modeling power, and modeling abstraction. Two simulation techniques are considered—Arena and SPN. Figure 10 illustrates the criteria for the evaluation of the two. The decision makers, according to linguistic terms, perform assessments of the weight of the criteria and ratings of the simulation techniques. The assessment matrix of the criteria and simulation technique alternatives is created by using a scale from 1 to 5, where 1 denotes worse and 5, best. The aggregate matrix for the weights of the simulation technique alternatives is computed using Eq. (16) and summarized in Table 5. Next, the best  $f_b^*$  and the worst  $f_b^-$  values of all criteria are formed by using Eqs. (2) and (3). Then, the S and R values are calculated using Eqs. (4) and (5), as listed in Table 6. The Q values for all simulation techniques calculated using Eq. (6), considering the maximum group utility,  $v=0.5$ , are summarized in Table 6. The simulation techniques are ranked based on S, R, and Q in descending order. Based on  $Q_a$  values, the selection of the best simulation technique was performed and summarized in Table 7.

As mentioned in Section 2.3, ‘S’ denotes the positive ideal solution, ‘R’ represents the negative ideal solution, and ‘Q’ means the optimal compromise solution. Therefore, decision-making can be performed by ordering the Q values in descending order. According to the Q values, the ranking of the simulation technique alternatives in descending order is formed as SPN > Arena. Hence, the best alternative is found to be SPN. In addition, both conditions, C1 and C2, are achieved, which means that  $Q(\text{Arena}) - Q(\text{SPN}) \geq 1/2 - 1$ . Similarly, SPN is highly ranked by R and S, which can guarantee its decision-making stability.

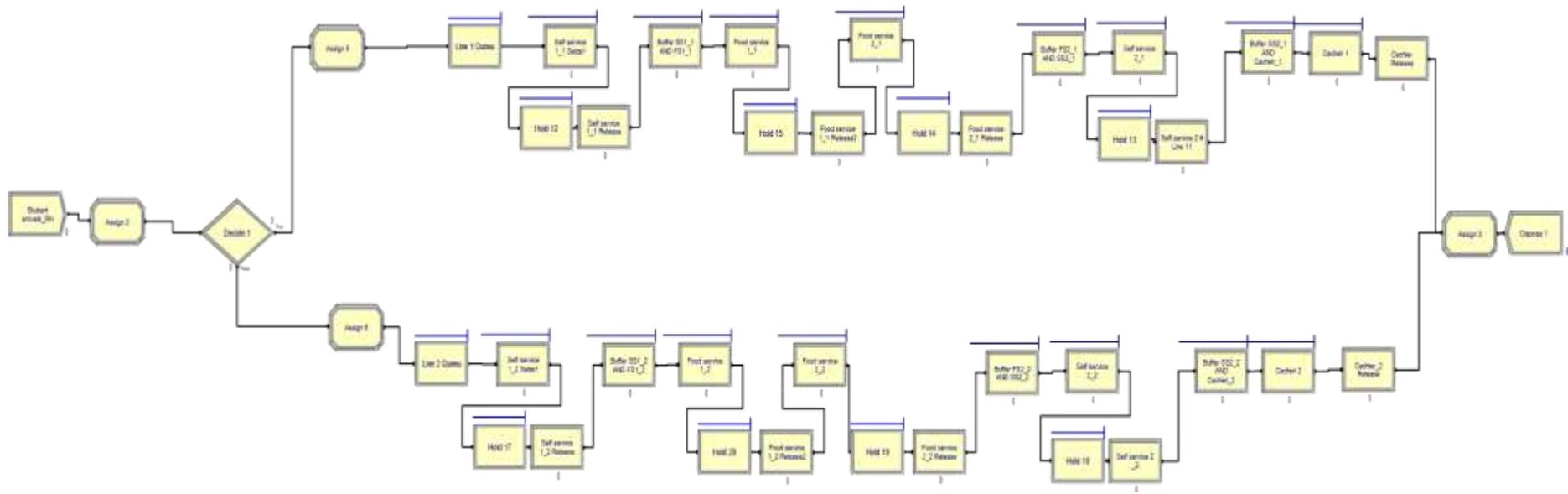


Figure 6. Arena model for system

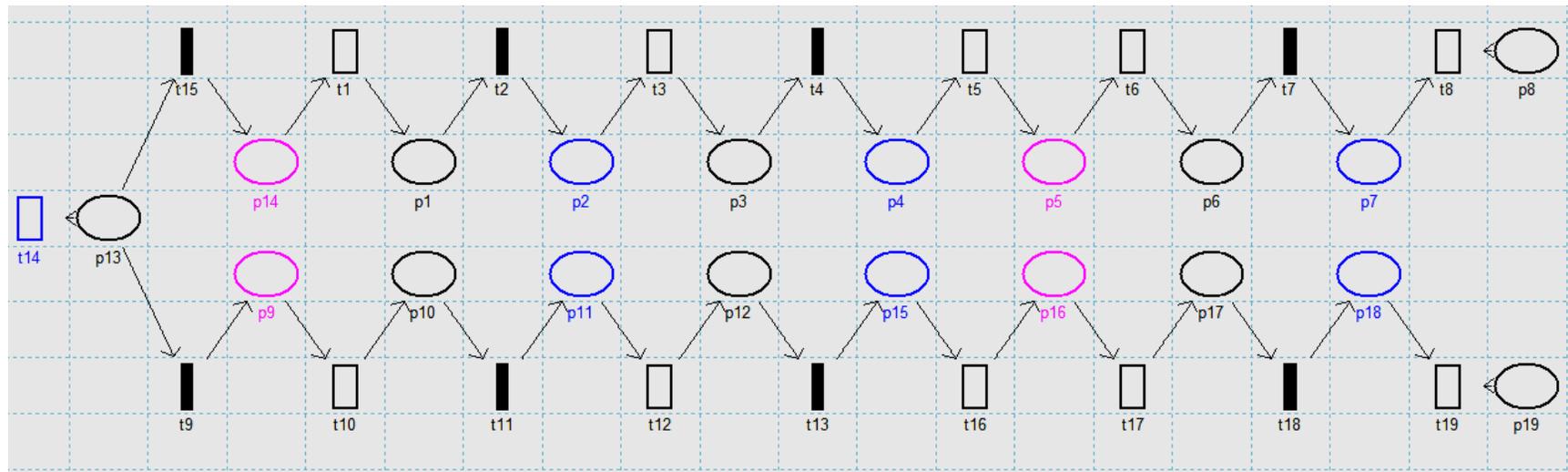


Figure 7. Stochastic Petri nets model for the system

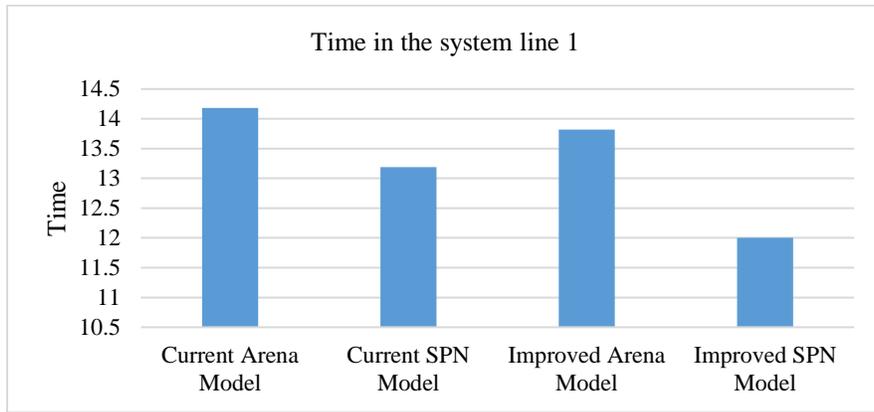


Figure 8. Time in the system line 1

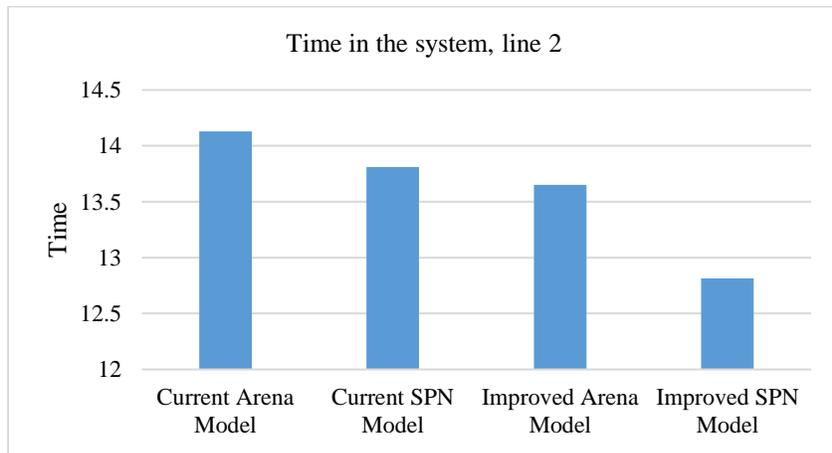


Figure 9. Time in the system line 2

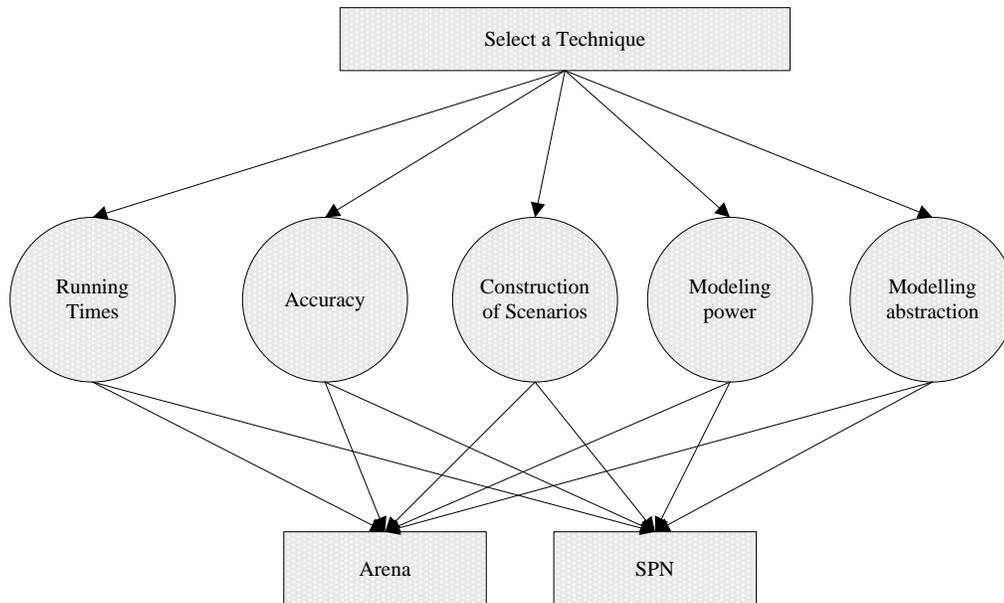


Figure 10. Criteria for the evaluation of techniques

**Table 5. Aggregate decision rating matrix for simulation techniques**

$C_{ij}$	Running Times	Accuracy	Construction of Scenarios	Modeling power	Modelling abstraction
Arena	3	3	3	5	4
SPN	5	4	4	1	5
Weight of criteria	3	5	4	4	5
$f_b^*$	5	4	4	5	5
$f_b^-$	3	3	3	1	4

**Table 6. Values of S, R and Q for simulation techniques**

$C_{ij}$	S	R	$Q_a$
Arena	17	5	1
SPN	4	4	0
	$S^- = 17$	$R^- = 5$	
	$S^* = 4$	$R^* = 4$	

**Table 7. Ranking of simulation techniques according to values of S, R, and Q.**

$C_{ij}$	S	Rank	R	Rank	$Q_a$	Rank
Arena	17	2	5	2	1	2
SPN	4	1	4	1	0	1

## 6. Conclusion

Simulation is the imitation of the operation of a system over time or real-world process. A descriptive technique of the simulation enables a decision-maker to evaluate the behavior of a model under various conditions. Two of the several different commercial off-the-shelf software—stochastic Petri nets and Arena—can be used to develop a simulation model. However, no comparative studies of these two formalisms have been conducted. Therefore, in this paper a methodology to evaluate the selected formalisms by using the ViseKriterijumska Optimizacija I Kompromisno Resenje (VIKOR) method is proposed. The criteria used to evaluate the best simulation technique are divided into running times, accuracy, development of scenarios, modeling power, and modeling abstraction. Two simulation techniques, Arena and SPN, are considered. Simulation models for the case study were developed using the Arena and SPN software, and the results show that long student waiting time exists at the KSU restaurant. Accordingly, in order to reduce such waiting time, one suggested improvement is made for this phase. Scenario changes are made to the self-service area by removing the second self-service section (Move the cold drinks to the first self-service section; move the food utensils and condiments to an area after the cashier section). As a result, the new averages of the waiting time in line 1 are 13.82 and 12.00145 min per student for Arena and SPN models, respectively. For line 2, the waiting time averages are 13.65 and 12.8154 min per student for Arena and SPN models, respectively. In addition, the results show that the Arena and stochastic Petri nets are convenient for modeling and analyzing the case study. However, based on the VIKOR approach, the comparative analysis shows that SPN is better than Arena. The future work on this study is to develop a flexible simulation model that can be applied to different types of systems that is able to yield more accurate results. In addition, more complex properties can be considered during the development of the model using the full version of the software. This will allow the creation of a more detailed model with a wider performance variety, which can be used to study resource allocations, such as the reduction of server idle time.

## Acknowledgements

## References

- Abdulaziz, M., Nasr, E. A., Al-Ahmari, A., Kaid, H. & Li, Z., Evaluation of Deadlock Control Designs in Automated Manufacturing Systems, International Conference on Industrial Engineering and Operations Management, 2015.
- Akman, G., Evaluating suppliers to include green supplier development programs via fuzzy c-means and VIKOR methods, *Computers & Industrial Engineering*, vol. 86, pp. 69–82, 2015.
- Al-Jaar, R. Y. & Desrochers, A. A., Performance evaluation of automated manufacturing systems using generalized stochastic Petri nets, *Robotics and Automation, IEEE Transactions on*, vol. 6, no. 6, pp. 621–639, 1990.
- Altiok, T. & Melamed, B. 2010. Simulation modeling and analysis with Arena, Academic press.
- Banks, J., CARSON II, J. S. & Barry, L. 2005. Discrete-event system simulation fourth edition, Pearson.
- Coman, D. & Ionescu, A., Simulation and Performance Analysis of a FMS/CIM Using Stochastic Timed Petri Nets, *Advanced Materials Research*, vol. 837, pp. 322–327, 2014.
- Deng, Y. & Chan, F. T., A new fuzzy dempster MCDM method and its application in supplier selection, *Expert Systems with Applications*, vol. 38, no. 8, pp. 9854–9861, 2011.
- Gallagher, A. G. & O’Sullivan, G. C. 2011. Simulations for Procedural Training. *Fundamentals of Surgical Simulation*. Springer.
- Kaid, H., El-Tamimi, A. M., Nasr, E. A. & Al-Ahmari, A. 2015. Applications of Petri nets Based Models in Manufacturing Systems: A Review. International Conference on Operations Excellence and Service Engineering. USA.
- Kang, D. & Park, Y., based measurement of customer satisfaction in mobile service: Sentiment analysis and VIKOR approach, *Expert Systems with Applications*, vol. 41, no. 4, pp. 1041–1050, 2014.
- Kelton, W. D., Sadowski, R. P. & Swets, N. B., *Simulation with Arena*.
- Law, A., *Simulation modeling and analysis*, McGraw-Hill series in industrial engineering and management science Show all parts in this series, 2007.
- Levine, A. I., DeMaria Jr, S., Schwartz, A. D. & Sim, A. J. 2013. The comprehensive textbook of healthcare simulation, Springer Science & Business Media.
- Li, Z. & Zhou, M., Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems, *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 34, no. 1, pp. 38–51, 2004.
- Liu, C., Shi, H. & Yuan, J., Modeling and performance analysis of flow lines with stochastic failures based on Petri nets, *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on*, pp. 6439–6444, 2008.
- Moore, K. E. & Gupta, S. M., Stochastic colored Petri net models of flexible manufacturing systems: Material handling systems and machining, *Computers & Industrial Engineering*, vol. 29, no. 1, pp. 333–337, 1995.
- Nasr, E. A., El-Tamimi, A. M., Al-Ahmari, A. & Kaid, H., Comparison and Evaluation of Deadlock Prevention Methods for Different Size Automated Manufacturing Systems, *Mathematical Problems in Engineering*, vol. 501, p. 537–893, 2015.
- Patel, A. M. & Joshi, A. Y., Modeling and analysis of a manufacturing system with deadlocks to generate the reachability tree using petri net system, *Procedia Engineering*, vol. 64, pp. 775–784, 2013.
- Petri, C. A. 1962. Kommunikation mit automaten. PHD Dissertation, University of Hamburg.
- Samkari, K. & Franz, V., A Petri net based method for the early verification & validation of a simulation study in construction management, *Proceedings of the Winter Simulation Conference*, p. 344, 2012.
- Sawhney, A., Petri Net based simulation of construction schedules, *Proceedings of the 29th conference on Winter simulation*, pp. 1111–1118, 1997.
- Stevenson, W. J. 2005. *Operations management*, McGraw-Hill.
- Tigane, S., Kahloul, L. & Bourekkache, S. 2017. Reconfigurable Stochastic Petri Nets for Reconfigurable Manufacturing Systems. *Service Orientation in Holonic and Multi-Agent Manufacturing*. Springer.
- Yan, H.-S., Wang, N.-S., Zhang, J.-G. & Cui, X.-Y., Modelling, scheduling and simulation of flexible manufacturing systems using extended stochastic high-level evaluation Petri nets, *Robotics and Computer-Integrated Manufacturing*, vol. 14, no. 2, pp. 121–140, 1998.
- Zhou, M., Modeling, analysis, simulation, scheduling, and control of semiconductor manufacturing systems: A Petri net approach, *IEEE Transactions on Semiconductor Manufacturing*, vol. 11, no. 3, pp. 333–357, 1998.
- Zhou, M., DiCesare, F. & Guo, D., Modeling and performance analysis of a resource-sharing manufacturing system using stochastic Petri nets, *Intelligent Control, 1990. Proceedings., 5th IEEE International Symposium on*, pp. 1005–1010, 1990.
- Zimmermann, A., Rodriguez, D. & Silva, M., A two phase optimization method for Petri net models of manufacturing systems, *Journal of Intelligent Manufacturing*, vol. 12, no. 5–6, pp. 409–420, 2001.