Journal of Theoretical and Applied Information Technology
15th June 2020. Vol.98. No 11

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

WEIGHTED ROUND ROBIN BASED PRODUCTION SCHEDULING MODEL FOR SEASONAL PRODUCT

PURBA DARU KUSUMA
School of Electrical Engineering, Telkom University, Indonesia
E-mail: purbodaru@telkomuniversity.ac.id

ABSTRACT

Single shipment scenario is adopted in most of production scheduling models. Besides, in general, next order will be executed after current order is executed completely. This approach works well for regular products. Unfortunately, for seasonal products, this approach triggers rejections or some customers may loose the peak season opportunity because they receive products after the season ends. Based on this problem, in this work, we propose new approach in production scheduling model based on Weighted Round Robin (WRR) scheduling method which is common in computer and networking fields. We propose two Round Robin variants: absolute Round Robin and relative Round Robin. This model is then simulated and compared with conventional First Come First Served (FCFS) model in production process environment for seasonal product. The adjusted parameters are: average inter-arrival date, average order quantity, and number of machines. Meanwhile, the observed parameters are: average waiting time, average completion time, and average flow time. Based on the simulation result, both absolute Round Robin and relative Round Robin produce lower waiting time rather than the conventional FCFS so that all customers can benefit in preparing and maximizing the peak season opportunity. Meanwhile, these both Round Robin models produce longer completion time. The relative Round Robin performs better than absolute Round Robin in creating lower completion time and lower flow time. Because the order is delivered in multiple shipments, this work shows that by using these proposed Round Robin based scheduling models, customer can benefit wider financial space rather than by using conventional First Come First Served Model in handling seasonal product.

Keywords: *Order fulfillment, Scheduling, Weighted Round Robin, Seasonal Product.*

1. INTRODUCTION

Scheduling is important aspect in production process. It is a part of decision making process with goal to optimize several objectives [1]. Company or factory must determine proper scheduling management so that it can execute its accepted orders by using its machines or resources. Besides in manufacture, scheduling is also important in transportation, service industry, or computing environment [1]. It is because company can only predict but cannot ensure the orders arrival time and quantity. Bad scheduling may affect several conditions: low troughput and tardiness in order completion that may end with penalty.

In the order completion mechanism, there is one general concept that is mostly adopted. Order is completed in single shipment so that in a single machine approach, orders will be executed sequentially [2]. The consequence is the next order will be executed after current order is executed completely. It makes higher prioritized orders will be executed earlier than lower prioritized orders. This approach is rationale because prioritization is related to maximizing profit and machine utilization and minimizing potential lost due to tardiness penalty.

Unfortunately, this approach creates problem in executing seasonal products. In seasonal products, all customers have same interest: receiving products as early as possible. Receiving the products earlier means customers can maximize sales and momentum because after the season period ends, demand for these products tends to be extreme low so that product must be sold in discounted price [3]. In sequential order execution where order is shipped in single shipment, there is potential where some customers receive products earlier so that they can maximize profit while other customers will loose momentum because they receive products later during the end of the season or after the season ends.

Based on this problem, we propose new production scheduling model. In this model, rather than adopt sequential scheduling and single shipment approaches, we adopt rotational and multi shipment approaches. It means single order will be splitted into several packets. In this work, we use Weighted Round Robin (WRR) method. Round Robin method is not popular in manufacture area. In the other side, this method is very popular in telecommunication and computer science areas, such as in data transmission [4] or server load balancing [5,6]. Based on this explanation, the novelty of this work is improving the Weighted Round Robin method which is popular in computer science to solve problem in manufacture field, especially in production scheduling for seasonal products. This work is also part of works in implementing computational solution in manufacture field.

The organization of this paper is as follows. The first section consists of background, research purpose, and the paper organization. The second section is the literature review which consists of review of works in production scheduling, seasonal product, and Round Robin method. The third section discusses the problem definition. The fourth section discusses the proposed model. The fifth section is the implementation of the proposed model into production process simulation application and testing result. The sixth section discusses the research findings, comparison between this work and the previous works, and the limitations of this work. The seventh section consists of conclusion and future research potentials.

2. RELATED WORK

2.1 Common Scheduling Methods in Production Process

In scheduling process, company focuses mostly in three aspects: turnaround, timeliness, and throughput [2]. Turnaround is time required to complete a job or a task [2]. Timeliness conforms about how a given job can be completed in a given deadline [2]. Throughput means the amount of jobs that can be executed completely during a fixed period of time [2]. Based on these main parameters, there are several derivative parameters used to measure scheduling or production process performance: completion time, flowtime, lateness, and tardiness [2]. Other common parameters are processing time, release date, due date, and weight [1].

In external part, flexibility may affect the scheduling process, especially in regular contract. Flexibility becomes common because of the uncertainty in technology process, market demand, and price [7]. Flexibility may benefit both customers and manufacturers in avoiding penalty because of some reasons, such as tardiness, quantity reduction in delivery because of capacity shortage or missed forecasting or event quantity addition because of customers (retailers) increase their order quantity. Flexibility also benefits both parties from under-stocking and/or over-stocking [8]. Most common flexibility is minimum order flexibility [7][9][10]. Term variation rate is used in flexibility to determine how far flexibility can be given [9]. Variation rate is the percentage or size that is still tolerated from the ordered quantity.

To meet this goal, many scheduling methods have been developed, implemented, and modified. Some of them are: shortest processing time (SPT), longest processing time (LPT), weighted shortest processing time (WSPT), earliest due date (EDD), shortest processing time among available jobs (SPTA), preemptive SPT (PSPT), and preemptive EDD (PEDD) [11]. Besides, several other methods includes: first come first served (FCFS), last come first served (LCFS), lowest remaining number of operations (LRNOP), most remaining number of operations (MRNOP), shortest remaining processing time (SRPT), longest remaining processing time (LRPT), and service in random order (SIRO) [12,13].

In general, these scheduling methods are designed to maximize one parameter while ignoring other parameters. For example, earliest due date (EDD) is designed to minimize tardiness penalty risk. But, the fairness of the system will be questioned because the earlier arriving jobs may be executed later because their due date is longer. In the other side, in first come first served method, the system or company is looked fair although tardiness penalty risk may increase. Meanwhile, SPT is designed to maximize the throughput because by prioritizing shorter processing time jobs, more jobs can be executed in a period of time. If a job represents a client or customer, completing more jobs means satisfying more customers.

Besides these common methods, several popular methods in computational science are also used in scheduling process, such as: genetic algorithm, ant colony optimization (ACO), agent based procedure, and machine learning [1]. In genetic algorithm, jobs are viewed as members in a population [1]. In machine learning based scheduling process, solutions in the past are used to create solution for future condition. Agent based scheduling model has been used by IBM in its scheduling software [1].

2.2 Seasonal Product

Seasonal product can be defined as product which its availability and sales time interval is limited [3]. It makes seasonal product price is higher when its sales period comes [3]. It is because demand for seasonal product tends to be high during the season period. Meanwhile, its price is lower outside its season period [8]. It is different with regular product where its price and availability tends to be stable over time.

A product is seasonal because of some reasons or determinants. The first determinant is weather [14,15]. Weather affects demand for products, such as ice cream, cold drink, outwear, and fashion [14]. The second determinant is event [15,16]. For example, demand for christmast tree is high every December and demand for dates is high during Ramadhan. Demand for flower, especially rose is high during February. The third determinant is product lifecycle [17]. Products with short lifecycle, such as mobile phone or computer can be seasonal too. Innovation makes the product lifecycle is short, especially for high-tech product. A new type of smartphone should be sold in a short period of time until other new smartphone comes. After new smartphone is released, customer tends to purchase the new version rather than the previous ones. It means producer can sell smartphone in premium price only during its first release period. After that, the price is lower. Margeson also noted that in semiconductor industry, a semiconductor generation lifecycle is only 6 months [17].

Handling seasonal products order is different from regular one. In seasonal product, order will be high when its season comes and will be low outside its season. Meanwhile, the production level is restricted because of resource limitations, such as raw material, machine, people, and inventory capacity [18]. It means, during the peak season, company may face order rejection potentials because its limited resource cannot meet the orders. Company can also face revenue potential lost due to this rejection [7]. Meanwhile, in the low season, company may face lower resource utility. In some case, the long lead time occurs [19]. Seasonal product also may trigger stock-out cost [15].

In many researches, this problem is solved by focusing in two aspects: production planning and inventory management. In the beginning, every company should make demand forecasting for its seasonal product [17,20]. Yenradee proposed yearly demand forecasting based on product group [20]. Then, the product group forecasting is detailed into individual product forecasting [20]. Margeson proposed combined model between short and medium-term forecast and exponential smoothing [17]. His work was simulated in stochastic lead time and demand patterns [17]. Inventory management is also important because in one side, company should have safety stock while higher safety stock means higher holding cost [20]. Meanwhile, increasing production is also limited by machine capacity and inventory capacity [20]. The other strategy is initiating production earlier and then persuading customers to hold this product by giving incentive for compensation [19].

2.3 Round-Robin Scheduling Model

In general, Round-Robin scheduling is a derivative of the first come first served (FCFS) scheduling [21]. It means that task that comes earlier will be executed earlier too. The difference is in basic FCFS method, current task must be executed completely first before next task is executed. Meanwhile, in Round-Robin scheduling, a task is splitted into several slices. After executing a slice of task completely, system will execute a slice of next task. The motivation is increasing fairness in the system so that the waiting time is reduced. In Round-Robin scheduling, later task may be finished earlier than earlier task because its size is smaller than the earlier task. In CPU process, the slice size is determined by the clock tick or tick timer [21]. This time slice is also called time quantum [22]. This mechanism makes Round-Robin method become one of the most efficient and effective technique in CPU scheduling [22].

This basic characteristic makes Round-Robin method is used widely in computer technology, in load balancing or resource allocating in parallel or cloud system. In cloud computing environment where there are several computing resources run or execute applications or requests over the internet, resource allocation can be defined as assigning requests or applications to available resources [23]. Meanwhile, load balancing is part of resource allocation process in cloud computing in order to improve the cloud performance by balancing the load among resources [6].

Round-Robin is also common used in scheduling and or load balancing process in computer network [5] or telecommunication network [4]. In computer network, there are many devices that are connected to each others, such as database servers, web servers, hosting, internet access, proxy, switches, routers, data center, etc [5]. In their work, Round-Robin is implemented as load balancer between requests and servers which means to dispatch incoming requests to certain available server [5]. In telecommunication network, Round-Robin can be used to manage the packet data flow, especially in queuing process that handles several connections [4]. In telecommunication network, Round Robin as one load balancing method is also important to maintain the network flexibility [24]. In telecommunication network, other observed aspects in scheduling process are fairness and throughput [24].

Based on this review, there is opportunity to improve production scheduling for seasonal product. Conventional FCFS model is fair enough for regular production process because earlier orders will be prioritized than the later orders. Unfortunately, this method may cause problem for seasonal products because in seasonal products, time space for selling products in premium price is limited. Customers that order products later may lose this opportunity. This FCFS model also may triggers condition when several customers order products in huge quantity so that they can monopolize resources. There is opportunity for adopting Round Robin scheduling model to solve this problem, Although it is not common in manufacturing field, it is popular in computer and networking fields. By splitting process into several processes, the main advantage of the Round Robin model is reducing waiting time so that all jobs have equal opportunity in using resources.

3. PROBLEM DEFINITION

Many companies have strong corporate brand or product brands. In some cases, a clothing company launches new seasonal product, for example Moslem apparel during Eid holiday. Three months before this holiday season, this company distributes the catalogue so that retailers can order the product. The company opens the order window for this product for two months so that it has time space for production. Then, the orders may come in various times. The order list that comes during the order window is presented in Table 1. Each order represents a retailer.

This company then produces the product based on orders in Table I. Its production capacity is 50 units per day. Orders are executed based on first come first served (FCFS) method. Each order will be delivered in a single shipment. The next order will be executed after current order is executed completely. The production time and the delivery time are also presented in Table 1.

Based on data in Table 1, it is presented that the first six retailers can secure the Eid holiday season completely because their order arrives before the season starts. The first retailer takes the highest benefit because their order arrives the earliest so that it can prepare the season better. Meanwhile, the seventh retailers still benefits the season although the reveue is not optimum due to opportunity lost in the first six days.

Table 1. Order List

Order ID	Arrival Time (day)	Qty (unit)	Proc. Time (days)	Del. Time (days)
1	1st	500	10	10th
2	3rd	700	14	24th
3	4th	650	13	37th
4	6th	850	17	54th
5	10th	450	9	63rd
6	15th	1,000	20	83rd
7	20th	750	15	98th
8	25th	800	16	114th

The 8th retailer gets the worst condition because its order arrives after the Eid season ends. It does not enjoy the highest demand and premium price. It makes this 8th retailer must sell its product at discounted price. In many production planning models, the 8th order is recommended to be rejected because company cannot complete this 8th order before the season starts.

4. PROPOSED MODEL

Based on this problem, we propose new order fulillment or production scheduling model based on Round Robin model. It means that company can executed other orders before current order is executed completely. In this model, it is assumed that due date or tardiness penalty is not applied. In other word, time flexibility is applied. Orders also can be delivered in more than one shipment. In this work, we use some notations for

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

variables. The explanation of the notations is as follows. The Round Robin production scheduling illustration is presented in Figure 1.

n_o = number of orders
n_{ao} = number of unfinished orders
n_m = number of machines
o_a = unfinished order
O_a = set of unfinished orders
n_b = block size
n_s = shipment size
i = index of the block in an order
j = index of order
$b_{i,j}$ = block in the system
$s_{i,j}$ = status of the block
r = round
b_r = block in a round
l = block index
R = set of rounds
n_{btr} = total number of blockes in a round
n_{br} = number of blockes of an order in a round
n_u = number of unexecuted blockes in an order
n_{TB} = block threshold
$T_{m,t}$ = machine toggle at time t

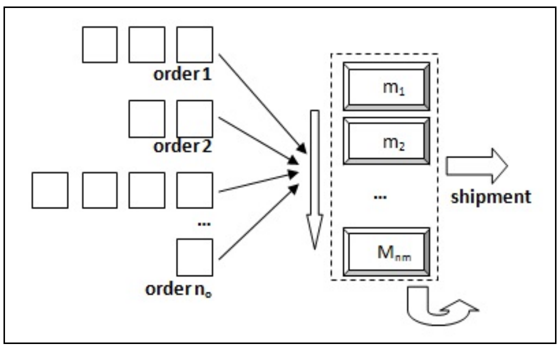

Figure 1. Round-Robin Scheduling Illustration

In this Round Robin production scheduling, orders is grouped into several rounds. A round must be executed first before production process executes the next round. Term executed means that all orders in a round have entered the production machines. A single round must consists of at least one block from every unfinished order. After all blocks in a round are executed then next round can be created. When a new order comes during the execution of current round, this new order is not involved in current round but will be involved in the next round.

The main algorithm of this Round Robin production scheduling is presented in Figure 2. In Figure 2, there are three subprograms: create_round, execute, and calculate_activeorder. The create_round function is used to create a round. The execute procedure is used to allocate blocks in the current round into production machines. The calculate_activeorder function is used to calculate how many active or unfinished orders are still in the system.

```
begin
 k ← 1
 while nao > 0 do
 begin
  rk ← create_round(Ao)
  execute(rk)
  k++
  nao ← calculate_activeorder(O)
 end
end
```

Figure 2. Round Robin Production Scheduling Main Algorithm

In the round creation process, in this work, weighted Round Robin (WRR) is adopted. As it is mentioned, there must be at least a block from every unfinished order in a single sound. Meanwhile, a round may consists of more than one block from an unfinished order. This concept is adopted based on WRR in data communication [4] where higher bandwidth queue may transmit more than one packet each time that it is visited.

In this work, we propose two methods in determining number of blockes of every order that can be executed in a single round. The first method is absolute method. The second method is relative method.

In the absolute method, the number of blockes that is executed in a single round is not affected by other orders. Formalization of the absolute method is presented in Equation 1 and Equation 2.

$$n_{br,j,k} = \mathrm{int}\left(\frac{n_{u,j}}{n_{Tb}}\right) \quad (1)$$

$$n_{br,j,k} = \begin{cases} n_{br,j,k}+1, n_{br,j,k} < 1 \\ n_{br,j,k}, n_{br,j,k} \geq 1 \end{cases} \quad (2)$$

Illustration of this absolute method is as follows. Assumed that there are three unfinished orders which the number of unexecuted blockes is {4, 6, 2}. If the block threshold is 2 then the number of blockes in a round is {2, 3, 1}. Meanwhile, if the block threshold is 4 then the number of blockes in a round is {1, 2, 1}. Based on this illustration, it is shown that in a round, there is

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

at least one block allocated in a round. It is also shown that order which has more unexecuted blockes has bigger opportunity to allocate more blockes in a single round.

In the relative method, the number of blockes of an order that is executed in a single round is affected by other orders. Formalization of the relative method is presented in Equation 3 to Equation 4. Result from Equation 4 is then processed by using Equation 2 to avoid zero value which means an unfinished order is fail to submit at least a block.

$$n_{\min U} = \min(N_u) \tag{3}$$

$$n_{br,j,k} = \text{int}\left(\frac{n_{u,j}}{n_{\min U}}\right) \tag{4}$$

Example in the absolute method can be used to illustrate the relative method. By using this data, the minimum number of unexecuted blockes is 2. By using Equation 4, the number of blockes in a round is {2, 3, 1}. Meanwhile, if the number of unexecuted blockes is {4, 6, 3} then the number of blockes in a round is {1, 2, 1} because the minimum number of unexecuted blockes is 3.

The queue of blockes in a single round is arranged based on the the order index. Blockes from an order is arranged consecutively. The blockes arrangement algorithm is presented in Figure 3.

```
begin
 l ← 1
 for j = 1 to nao with step = 1 do
 begin
  for i = 1 to nbr,j,k with step = 1
do
  begin
   br,l ← bi,j
   l++
  end for
  nu,j ← nu,j – nbr,j,k
 end for
end
```

Figure 3. Blockes Arrangement Algorithm

The last process is allocating blocks into machines. In this work, the processing time of every block is equal. So, basic Round Robin method is applied in this process. This process is formalized in Equation 5. Meanwhile, the block to machine allocation algorithm is presented in Figure 4. In Figure 4, variable T is same with $T_{m,t}$.

$$T_{m,t} = \begin{cases} T_{m,t-1}+1, T_{m,t-1} < n_m \\ 1, T_{m,t-1} = n_m \end{cases} \tag{5}$$

```
begin
 for l = 1 to nbtr with step = 1 do
  allocate(br,l,mT)

  if T < nm then
   T ← T + 1
  else
   T ← 0
 end for
end
```

Figure 4. Block to Machine Algorithm

5. IMPLEMENTATION AND TESTING

This proposed model then is implemented into production simulation application. This application is developed by using PHP languange. The environment of this production simulation is a factory that consists of several machines. Each machine has production capacity 50 units per day. The order opening period is 60 days. Meanwhile, the production period is 200 days. The order opening starting day is same day with the production period starting day.

The evaluation criteria or parameters that are used in this works are as follows. Parameters are divided into two groups: adjusted parameters and observed parameters. These parameters are common used in many works or researches in production planning or scheduling. There are three adjusted parameters: average inter-arrival date (t_{avarr}), average order quantity (n_{avo}), and the number of machines (n_m). The inter arrival date and order quantity is generated randomly and follows exponential distribution. It is because in the real world, orders arrival usually follows Poison process. There are three observed parameters: average waiting time (t_{avwait}), average completion time (t_{avcomp}), and average flow time (t_{avflo}) as they are common observed parameters in production system [2]. These aspects are the main parameters in evaluating production planning or scheduling performance.

In this simulation, we compare the performance of the proposed models (absolute Round Robin and relative Round Robin) and the common first come first served (FCFS) scheduling model. These comparison parameters are formalized by using Equation 6 to Equation 11.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$r_{wait\_abFCFS} = \frac{t_{avwait\_abRR}}{t_{avwait\_FCFS}} \quad (6)$$

$$r_{wait\_relFCFS} = \frac{t_{avwait\_relRR}}{t_{avwait\_FCFS}} \quad (7)$$

$$r_{comp\_abFCFS} = \frac{t_{avcomp\_abRR}}{t_{avcomp\_FCFS}} \quad (8)$$

$$r_{comp\_relFCFS} = \frac{t_{avcomp\_relRR}}{t_{avcomp\_FCFS}} \quad (9)$$

$$r_{flo\_abFCFS} = \frac{t_{avflo\_abRR}}{t_{avflo\_FCFS}} \quad (10)$$

$$r_{flo\_relFCFS} = \frac{t_{avflo\_relRR}}{t_{avflo\_FCFS}} \quad (11)$$

Definition of variables in Equation 6 to Equation 11 is as follows. The r_{wait_abFCFS} is waiting time ratio between absolute Round Robin and FCFS. The $r_{wait\_relFCFS}$ is waiting time ratio between relative Round Robin and FCFS. The r_{comp_abFCFS} is completion time ratio between absolute Round Robin and FCFS. The $r_{comp\_relFCFS}$ is completion time ratio between relative Round Robin and FCFS. The r_{flo_abFCFS} is flow time ratio between absolute Round Robin and FCFS. The $r_{flow\_relFCFS}$ is flow time ratio between relative Round Robin and FCFS. The t_{avwait_abRR} is the average waiting time of absolute Round Robin. The t_{avwait_relRR} is the average waiting time of relative Round Robin. The t_{avwait_FCFS} is the average waiting time of FCFS. The t_{avcomp_abRR} is the average completion time of absolute Round Robin. The t_{avcomp_relRR} is the average completion time of relative Round Robin. The t_{avcomp_FCFS} is the average completion time of FCFS. The t_{avflo_abRR} is the average flow time of absolute Round Robin. The t_{avflo_relRR} is the average flow time of relative Round Robin. The t_{avflo_FCFS} is the average flow time of FCFS.

The first test group is analyzing the relation between the average inter arrival date and the observed parameters. The inter arrival date ranges from 1 to 10 days with step size is 1 day. There are five simulation sessions for every average inter-arrival date. In this test group, the average order quantity is set 1000 units. There are five machines in the system. The result is presented in Table 3 for average waiting time, Table 3 for average completion time, and Table 4 for average flow time.

Table 2. Relation Between Average Inter Arrival Date and Average Waiting Time

t_{avarr} (days)	t_{avwait} (days)		
	FCFS	Absolute RR	Relative RR
1	17.57	7.30	2.34
2	5.24	2.09	0.49
3	1.12	0.35	0.03
4	2.24	0.85	0.10
5	0.42	0.12	0.01
6	1.06	0.38	0.03
7	0.31	0.10	0.00
8	0.98	0.29	0.00
9	1.12	0.43	0.00
10	0.26	0.04	0.00

Table 2 presents the negative relation between average inter-arrival date and average waiting time. FCFS performs the worst model in producing lower waiting time. Disparity with the proposed models is also wide. Meanwhile, relative Round Robin performs better than absolute Round Robin in average waiting time aspect. It means that by using relative Round Robin scheduling model, orders tend to wait the shortest rather than by using other models.

Disparity comparison between the proposed models and the FCFS model in average waiting time due to the increasing of the average inter arrival date is as follows. The r_{wait_abFCFS} is 0.42 when the t_{avarr} is 1 day and it falls to 0.16 when the t_{avarr} is 10 days. The $r_{wait\_relFCFS}$ is 0.13 when the t_{avarr} is 1 day and it falls to 0 when the t_{avarr} is 10 days. It means that disparity between proposed models and FCFS model is wider due to the increasing of the inter arrival date.

Table 3 presents the negative relation between average inter-arrival date and average completion time. Comparing between FCFS and the proposed models, the completion time in proposed models is significant higher than the FCFS model. Meanwhile, due to the increasing of the inter-arrivale date, the disparity is reduced fast. Comparing between the absolute Round Robin and relative Round Robin, relative Round Robin performs better than absolute Round Robin in the average completion time aspect. It means that the processing time in a factory tends to be the fastest while the system adopts FCFS model.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 3. Relation Between Average Inter Arrival Date and Average Completion Time

t_{avarr} (days)	t_{avcomp} (days)		
	FCFS	Absolute RR	Relative RR
1	1.68	23.26	17.55
2	1.53	7.96	5.75
3	1.36	2.82	2.47
4	1.74	4.22	3.77
5	1.25	1.61	1.47
6	1.36	2.18	1.98
7	1.67	2.04	1.85
8	2.20	3.53	3.40
9	1.93	2.88	2.61
10	1.12	1.49	1.40

Disparity comparison among models in average completion time due to average inter arrival date is as follows. The average completion time ratio between absolute Round Robin and FCFS is 13.83 when the inter arrival date is 1 day and it falls to 1.33 when the inter arrival date is 10 days. Meanwhile, the average completion time ratio between relative Round Robin and FCFS is is 10.43 when the average inter arrival date is 1 day and it falls to 1.25 when the average inter arrival date is 10 days. It means that disparity is wide when the intar arrival date is low and it is narrow when the inter arrival date is high.

Table 4. Relation Between Average Inter Arrival Date and Average Flow Time

t_{avarr} (days)	t_{avflo} (days)		
	FCFS	Absolute RR	Relative RR
1	19.26	30.56	19.89
2	6.77	10.05	6.24
3	2.48	3.17	2.50
4	3.98	5.07	3.87
5	1.67	1.73	1.48
6	2.42	2.56	2.01
7	1.98	2.14	1.85
8	3.19	3.82	3.40
9	3.04	3.30	2.61
10	1.38	1.53	1.40

Table 4 presents the negative relation between average inter-arrival date and average flow time. In general, FCFS model performance is similar to the relative Round Robin aspect in flow time aspect. Meanwhile, the absolute Round Robin performs the worst in the flow time aspect. It means that in factory, product lifetime tends to be equal between FCFS model and relative Round Robin model.

Average flow time disparity among models due to the increasing of inter arrival date is as follows. The r_{flo_abFCFS} is 1.587 when the t_{avarr} is 1 day and it falls to 1.113 when the t_{avarr} is 10 days. It means that the average flow time disparity between absolute Round Robin and FCFS is narrowing due to the increasing of the average inter arrival date. Meanwhile, the $r_{flo\_relFCFS}$ tends to fluctuate from 0.831 to 1.066. It means that the inter arrival date does not affect the average flow time disparity between relative Round Robin and FCFS.

The second test group is analizing relation between average order quantity and the observed parameters. In this test group, the average order quantity ranges from 100 to 1,500 units with step size is 100 units. During this test, the average inter-arrival date is set 5 days. There are five machines in the factory. There are five simulation sessions in every average order quantity. The result is presented in Table 5 for average waiting time, in Table 6 for average completion time, and in Table 7 for average flow time.

Table 5. Relation Between Average Order Quantity and Average Waiting Time

n_{avo} (units)	t_{avwait} (days)		
	FCFS	Absolute RR	Relative RR
100	0.00	0.00	0.00
200	0.00	0.00	0.00
300	0.16	0.05	0.00
400	0.50	0.18	0.03
500	0.48	0.11	0.00
600	0.64	0.20	0.00
700	0.79	0.28	0.02
800	5.43	2.12	0.35
900	2.25	0.85	0.01
1,000	8.61	3.50	0.33
1,100	5.94	2.50	0.24
1,200	7.75	3.19	0.18
1,300	10.66	4.46	0.42
1,400	11.16	4.72	0.43
1,500	17.38	7.45	0.42

Table 5 presents the positive relation between average order quantity and average waiting time. Comparing between FCFS model and the proposed models, the proposed models performs better in creating lower waiting time rather than FCFS model. The disparity is significant high due to the increasing of the average order quantity. Meanwhile, comparing between absolute Round Robin and relative Round Robin, the relative Round Robin performs also performs better in creating lower waiting time and the disparity is significant high due to the increasing of the average order quantity.

Disparity among models in average waiting time due to average order quantity is as follows. The r_{wait_abFCFS} is 0.328 when the n_{avo} is 300 units and it rises to 0.428 when the n_{avo} is 1,500 units. It means that the increasing of number of orders makes the average waiting time disparity between absolute Round Robin and FCFS wider. Meanwhile, the $r_{wait\_relFCFS}$ tends to fluctuate due to the increasing of n_{avo} from 0 to 0.064. It means that the average order quantity does not affect the average waiting time disparity between relative Round Robin and FCFS.

Table 6 presents the positive relation between average order quantity and average completion time. Comparing between the FCFS model and the proposed models, it is shown that the FCFS model performs better in creating lower average completion time. This disparity is significant high due to the increasing of the average order quantity. Comparing between the absolute Round Robin and the relative Round Robin, it is shown that relative Round Robin performs better than the absolute Round Robin.

Table 6. Relation Between Average Order Quantity and Average Completion Time

n_{avo} (units)	t_{avcomp} (days)		
	FCFS	Absolute RR	Relative RR
100	0.00	0.00	0.00
200	0.30	0.31	0.30
300	0.53	0.71	0.67
400	1.11	1.74	1.59
500	1.43	2.24	1.96
600	1.47	2.23	2.06
700	1.86	2.55	2.38
800	2.93	14.02	9.99
900	2.86	5.82	4.61
1,000	3.03	16.30	10.00
1,100	3.86	12.89	8.99
1,200	4.13	16.24	10.54
1,300	4.58	21.05	15.84
1,400	4.78	20.99	13.61
1,500	6.40	33.00	22.49

Average completion time disparity among models due to the increasing of average order quantity is as follows. The t_{avcomp_abFCFS} is 1.048 when the n_{avo} is 200 units and it rises up to 5.158 when the n_{avo} is 1,500 units. Meanwhile, the the $t_{avcomp\_relFCFS}$ is 1 when the n_{avo} is 200 units and it rises up to 3.515 when the the n_{avo} is 1,500 units. It means that the average completion time disparity between Round Robin and FCFS becomes wider due to the increasing of the average order quantity.

Table 7. Relation Between Average Order Quantity and Average Flow Time

n_{avo} (units)	t_{avflo} (days)		
	FCFS	Absolute RR	Relative RR
100	0.00	0.00	0.00
200	0.30	0.31	0.30
300	0.70	0.76	0.67
400	1.61	1.92	1.62
500	1.91	2.35	1.96
600	2.11	2.43	2.06
700	2.65	2.83	2.40
800	8.36	16.14	10.34
900	5.11	6.67	4.62
1,000	11.64	19.80	10.33
1,100	9.80	15.39	9.22
1,200	11.88	19.43	10.72
1,300	15.24	25.52	16.26
1,400	15.93	25.72	14.04
1,500	23.78	40.45	22.92

Table 7 presents the positive relation between average order quantity and average flow time. Comparing between FCFS model and the proposed models, FCFS model performs equal with the relative Round Robin model. Meanwhile, absolute Round Robin model performs the worst in creating low flow time. This disparity is high due to the increasing of the average order quantity.

The average flow time disparity among models due to the increasing of the average order quantity is as follows. The t_{avflo_abFCFS} is 1.048 when the n_{avo} is 200 units and it rises up to 1.701 when the n_{avo} is 1,500 units. It means the average flow time disparity between absolute Round Robin and FCFS is wider due to the increasing of the average number of orders. Meanwhile the $t_{avflo\_relFCFS}$ tends to fluctuate from 0.881 to 1.237. It means that the average order quantity does not affect the average flow time disparity between relative Round Robin and FCFS.

The third test group is analyzing the relation between number of machines and the observed parameters. In this scenario, the number of machines ranges from 3 to 15 units with the step size is 1 unit. Each machine has production capacity 50 units per day. Average order quantity is set 1000 units. Average inter-arrival date is set 3 days. There are five simulation sessions for every number of machines. The result is shown in Table 8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

for average waiting time, in Table 9 for average completion time, and in Table 10 for average flow time.

Table 8. Relation Between Number of Machines and Average Waiting Time

n_m (units)	t_{avwait} (days)		
	FCFS	Absolute RR	Relative RR
3	40.85	17.18	2.62
4	21.60	9.07	0.98
5	12.30	4.96	0.71
6	8.09	3.37	0.29
7	13.12	5.40	0.50
8	7.22	2.80	0.31
9	2.79	1.09	0.01
10	3.35	1.31	0.04
11	2.49	0.98	0.03
12	0.95	0.31	0.00
13	0.65	0.18	0.00
14	0.34	0.07	0.00
15	0.18	0.03	0.00

In Table 8, it is presented that the number of machines has negative relation with the average waiting time. The average waiting time decreases due to the increasing of the number of machines. In the beginning, the average waiting time decreases fast. Then, the decreasing speed gets slower. The, after certain number of machines, the increasing of the number of machines does not affect the decreasing of the average waiting time.

Comparing among models, FCFS produces the worst average waiting time. It means, due to changing in number of machines, both proposed models still perform better than the conventional FCFS model. Meanwhile, comparing between proposed models, the relative Round Robin performs better than absolute Round Robin in creating low average waiting time.

Disparity in average waiting time among models changes due to the increasing of the number of machines. When there are few machines, this disparity is low. In the other side, when there are many machines, this disparity is high.

In Figure 5, there are two lines. The blue line indicates the r_{wait_abFCFS}. Meanwhile, the red line indicates $r_{wait\_relFCFS}$. Lower ratio means wider disparity. In Figure 5, it is shown clearly that in any number of machines, both proposed models produces lower average waiting time rather than conventional FCFS model because all ratios is smaller than 1. It is also shown that the relative Round Robin performs better than the absolute Round Robin significantly.

Figure 5 also presents the trend. In average waiting time ratio between absolute Round Robin and FCFS model, in the beginning, the lines falls slowly. After passing certain number of machines, the line falls fast. Meanwhile, in average waiting time ratio between relative Round-Robin and FCFS model, the line falls slowly, after passing certain number of machines, the ratio is stagnant in 0 because the average waiting time of relative Round Robin model is 0.

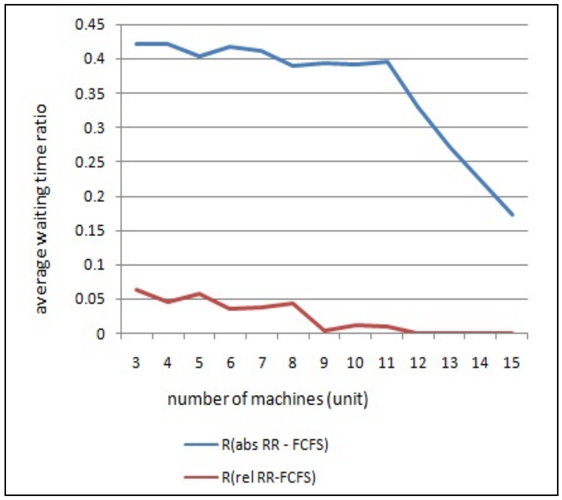

Figure 5. Relation between Number of Machines and Average Waiting Time Ratio

Table 9. Relation Between Number of Machines and Average Completion Time

n_m (units)	t_{avcomp} (days)		
	FCFS	Absolute RR	Relative RR
3	5.85	57.20	38.76
4	5.18	36.83	24.69
5	3.25	21.57	14.39
6	2.95	12.92	10.28
7	2.68	22.58	14.40
8	1.98	13.62	8.93
9	1.84	6.32	4.06
10	1.75	6.73	4.47
11	1.62	5.79	4.08
12	1.27	2.90	2.25
13	1.14	2.09	1.58
14	0.86	1.37	1.12
15	0.94	1.37	1.13

In Table 9, it is presented that there is negative relation between the number of machines and the average completion time. Average completion time increases due to the increasing of the number of machines. It is rationale because the increasing of machines means the the increasing of the total production capacity.

Comparing among models, it is shown that the conventional FCFS model still performs the best mode in creating low average completion time. Meanwhile, comparing between proposed models, the relative Round Robin model creates lower completion time rather than the absolute Round Robin model.

Disparity among models in average completion time decreases due to the increasing of the number of machines. This disparity is wide when there are few machines. In the other side, disparity is narrow when there are many machines. The r_{avcomp_abFCFS} is 9.78 when the n_m is 3 units and it falls to 1.46 when the n_m is 15 units. Meanwhile, the $r_{avcomp\_relFCFS}$ is 6.63 when the n_m is 3 units and it falls to 1.20 when the n_m is 15 units.

Table 10. Relation Between Number of Machines and Average Flow Time

n_m (units)	t_{avflo} (days)		
	FCFS	Absolute RR	Relative RR
3	46.70	74.38	41.39
4	26.78	45.90	25.67
5	15.55	26.53	15.10
6	11.03	16.30	10.57
7	15.80	27.98	14.90
8	9.20	16.42	9.25
9	4.63	7.41	4.07
10	5.09	8.04	4.51
11	4.11	6.77	4.11
12	2.21	3.21	2.25
13	1.79	2.27	1.58
14	1.19	1.45	1.12
15	1.12	1.40	1.13

Table 10 presents the negative relation between number of machines and the average flow time. The average flow time decreases due to the increasing of the number of machines. Comparing among models, FCFS performs the best in creating low average flow time. In the other side, absolute Round Robin creates the highest average flow time.

Disparity among models in average flow time due to the number of machines is as follows. The r_{avflo_abFCFS} is 1.59 when the n_m is 3 units and it falls to 1.25 when the n_m is 15 units. Meanwhile, disparity in average flow time between the relative Round Robin and FCFS is very low. The $r_{avflo\_relFCFS}$ fluctuates due to the increasing of the n_m. It means that the number of machines does not affect the disparity in average flow time between relative Round Robin and FCFS.

6. DISCUSSION

Based on this result, the findings are as follows. The first finding is that both absolute Round Robin and relative Round Robin performs better in creating lower waiting time so that orders can enter the production list earlier than the FCFS model as it is common production scheduling [11,12]. It is because in Round Robin model, later orders can be executed without waiting earlier orders are completed. In seasonal product, this condition is good because later orders do not need to wait to be executed rather than it happens in conventional order scheduling model which implements single shipment completion. So, many more customers can benefit high demand during the season. Meanwhile, the relative Round Robin performs better than absolute Round Robin in waiting time aspect.

The second finding is that the completion time in Round Robin tends to be longer than in FCFS model. It is because in FCFS model, each time an order enters the production list, then all machines will be allocated to execute this order without interference from other orders. Meanwhile, in seasonal product, longer completion time has positive advantage in financial aspect. Because the bill usually must be paid after the shipment arrives, customers have more financial space because in this Round Robin model, an order can be delivered in multiple shipments. So, while the customers wait for the next shipment arrival, they can sell the earlier arrived products first although the demand may be not as high as during the season high demand.

The example is as follows. In Table 6, when the average order quantity is 1,000 units, the completion time for relative Round Robin is 10 days. If it is assumed that the products will be delivered in two shipments (the 5th day and the 10th day) and the production distribution is proportional, then the customer can reduce the stock of the first 500 units in five days before the next 500 unit shipment arrives.

Besides improvements and contributions, there are limitations in this work. First, in this work, the factory produces only one product. Second, the capacity of every machine is uniform. In the real world, a factory usually produces multi products. The installed machines are also various in capacity and performance. Third, in this work, time between orders is ignored. Fourth, in this work, there are not any penalties due to delivery tardiness.

7. CONCLUSION AND FUTURE WORK

Based on the simulation result analysis and the findings, for seasonal product, the Round Robin models perform better than the conventional model, such as FCFS model, especially in reducing waiting time aspect. It means that customers which their order comes later can also prepare and maximize the peak season period. In completion time aspect, the FCFS model performs better than the Round Robin model because by using Round Robin model, the completion time tends to be longer than the FCFS model. Fortunately, the order is delivered in multiple shipments in Round Robin model so that customers may benefits better financial space in paying the arrived shipments. Comparing between two proposed models, the relative Round Robin performs better than the absolute Round Robin.

Based on this work, there are several future research potentials. The first potential is improving this proposed model in multi items order scenario. The second potential is improving these proposed models where the due date and penalty tardiness are applied.

REFERENCES:

[1] M.L. Pinedo, *Scheduling: Theory, Algorithms, and Systems*, 5th ed., Springer, 2016.

[2] K.R. Baker and D. Trietsch, *Principles of Sequencing and Scheduling*, John Wiley and Sons, 2009.

[3] G.P. Soysal and L. Krishnamurthi, "Demand Dynamics in the Seasonal Goods Industry: An Empirical Analysis," *Marketing Science*, vol. 31(2), 2012, pp. 293-316.

[4] C. Semeria, *Supporting Differentiated Service Classes: Queue Scheduling Disciplines*, Juniper Networks, 2001, white paper.

[5] Y. Afrianto, H. Sukoco, S. Wahjuni, "Weighted Round Robin Load Balancer to Enhance Web Server Cluster in OpenFlow Networks," *Telkomnika*, vol. 16(3), 2018, pp. 1402-1408.

[6] P. Samal and P. Mishra, "Analysis of Variants in Round Robin Algorithms for Load Balancing in Cloud Computing," *International Journal of Computer Science and Information Technologies*, vol. 4(3), 2013, pp. 416-419.

[7] S.P. Sethi, H. Yan, H. Zhang, "Quantity Flexibility Contracts: Optimal Decision with Information Updates," *Decision Sciences*, vol. 35(4), 2004, pp. 691-712.

[8] W. Cai, L. Abdel-Malek, B. Hoseini, S.R. Dehkordi, "Flexible Contracting with Minimum Order Quantity Commitment and Expedited Delivery," Proceeding of 18th *International Working Seminar on Production Economics*, Innsbruck, February 2014.

[9] C. Chu, E.E. Longomo, X. Song, and D. Ouelhaj, "Flexibility Analysis on Supply Chain Contract using a Parametric Linear Programming Model," *Proceeding of 2014 IEEE Conference on Automation Science and Engineering*, Taipei, August 2014.

[10] Y. Bassok and R. Anupindi, "Analysis of Supply Contracts with Commitments and Flexibility," *Naval Research Logistics*, vol. 55(5), 2008, pp. 459-477.

[11] P. Kaminsky and D. Hochbaum, *Due Date Quotation Models and Algorithms*, University of California, September 2003.

[12] H.I. Koruca and E. Aydemir, "A Priority Rule Based Production Scheduling Module on Faborg-Sim Simulation Tool," *Gazi University Journal of Science*, vol. 27(4), 2014, pp. 1143-1155.

[13] E. Aydemir and H.I. Koruca, "A New Production Scheduling Module Using Priority-Rule Based Genetic Algorithm," *International Journal of Simulation Modelling*, vol. 14(3), 2015, pp. 450-462.

[14] H. Fu, W. Chen, H. Soon, and H. Peng, "Coordination of the Supply Chain of Seasonal Products with Buy-Back Contract under Weather-Sensitive Demand," *AMSE JOURNALS Series: Advances A*, vol 53(2), 2016, pp. 18-33.

[15] A. Regattieri, F. Bindi, F. Lolli, R. Manzini, and A. Persona, "Operating Strategies for Production Planning of Seasonal Products," *Proceeding of 19th International Conference on Production Research*, Valparaiso, 29 July 2007.

[16] S. Radas and S.M. Shugan, "Seasonal Marketing and Timing New Product Introductions," *Journal of Marketing Research*, vol. 35(2), 1998, pp. 296-315.

[17] W.D. Margeson, *A Forecasting and Inventory Model for Short Lifecycle Products with*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Seasonal Demand Patterns, Massachusetts Institute of Technology, June 2002, master thesis.

[18] R. Metters, "Production Planning with Stochastic Seasonal Demand and Capacitated Production," *IIE Transactions*, vol 29(11), 1997, pp. 1017-1029.

[19] K. Huang, C. Kuo, and M. Lu, "Wholesale Price Rebate vs. Capacity Expansion: The Optimal Strategy for Seasonal Products in a Supply Chain," *European Journal of Operation Research*, vol. 234(1), 2014, pp. 77-85.

[20] P. Yenradee, A. Pinnoi, and A. Charoenthavornying, "Demand Forecasting and Production Planning for Highly Seasonal Demand Situations: Case Study of a Pressure Container Factory," *ScienceAsia*, vol. 27, 2001, pp. 271-278.

[21] M.A. Mohammed, M. Abdulmajid, B.A. Mustafa, and R.F. Ghani, "Queueing Theory Study of Round Robin Versus Priority Dynamic Quantum Time Round Robin Scheduling Algorithm," *Proceeding of 4th International Conference on Software Engineering and Computer Systems (ICSECS)*, Kuantan, 19 August 2015.

[22] M.M. Tajwar, M.N. Pathan, L. Hussaini, and A. Abubakar, "CPU Scheduling with a Round Robin Algorithm Based on an Effective Time Slice," *J. Int. Process. Syst.*, vol. 13(4), 2017, pp. 941-950.

[23] P. Pradhan, P.K. Behera, and B.N.B. Ray, "Modified Round Robin Algorithm for Resource Allocation in Cloud Computing," *Procedia Computer Science*, vol. 85, 2016, pp. 878-890.

[24] D. Perdana, A.N. Sanyoto, and Y.G. Bisono, "Performance Evaluation and Comparison of Scheduling Algorithms on 5G Networks using Network Simulator," *International Journal of Computers Communication and Control*, vol. 14(4), 2019, pp. 530-539.