Multi-objective Batch Scheduling in Collaborative Multi-product Flow Shop System by using Non-dominated Sorting Genetic Algorithm

Purba Daru Kusuma
Computer Engineering, Faculty of Electrical Engineering
Telkom University, Bandung, Indonesia

***Abstract*—Batch scheduling is a well-known topic that has been studied widely with various objectives, methods, and circumstances. Unfortunately, batch scheduling in a collaborative flow shop system is still unexplored. All studies about batch scheduling that are found were in a single flow shop system where all arriving jobs come from single door. In a collaborative flow shop system, every flow shop handles its own customers although joint production among flow shops to improve efficiency is possible. This work aims to develop a novel batch scheduling model for a collaborative multi-product flow shop system. Its objective is to minimize make-span and total production cost. This model is developed by using non-dominated sorting genetic algorithm (NSGA II) which is proven in many multi objective optimization models. This model is then compared with the non-collaborative models which use NSGA II and adjacent pairwise interchange algorithm. Due to the simulation result, the proposed model performs better than the existing models in minimizing the make-span and total production cost. The make-span of the proposed model is 10 to 17 percent lower than the existing non-collaborative models. The total production cost of the proposed model is 0.3 to 3.5 percent lower than the existing non-collaborative models.**

Keywords—Batch scheduling; flow shop; NSGA II; collaborative system

I. Introduction

Batch scheduling is a well-known topic in supply chain management, especially in production process. In general, batching mechanism is grouping the jobs that must be processed based on their similarities so that each batch is processed together [1]. Although its concept is simple, there are many studies in batching process because the circumstances in the production system are complex and various, such as multi-site plant [2], multiple products [3,4], deteriorating jobs [5], limited batch size [6], parallel batching [7], serial batching [8,9], and so on. There is not any single model that is the best to solve all problems in batching process. Most studies in batching process focused on the batch scheduling [10,11] while others focused on determining the batch size [2] and the number of batches [2].

Batch scheduling studies were also studied in flow shop system. In a flow shop, production is divided into several stages [12]. These stages can be production, assembling, or inspection. The flow shop can be permutation or non-permutation. In the permutation flow shop, once these jobs are sequenced, this sequence will be fixed for all stages [13]. In the non-permutation flow shop, the jobs sequence among stages may be different [14]. In the batch scheduling in a flow shop system, the jobs are batched first before sequenced.

Like the flow shop scheduling problem, batch scheduling problem is also a combinatorial optimization problem. There are many methods or algorithms to solve the batch scheduling problem, such as genetic algorithm (GA) [8,13], mixed integer linear programming (MILP) [15], shortest processing time (SPT) [10], variable neighborhood descent (VND) [10], backward dynamic programming [5], memetic algorithm [16], back-off decomposition algorithm [17], longest due date (LDD) [9], adjacent pairwise interchange (API) [9], permutation method (PM) [9], and so on. There are several parameters that were used as objective in a batch scheduling studies. The most common parameter is make-span [14,15]. Other parameters are total actual flow time (TAFT) [9,10], total tardiness [18], total earliness [18], revenue [17], overstocking [16], delay [16], inefficiency cost [16], total completion time [5,12], total energy cost [19], number of late jobs [6], run time [20], and so on.

The problem in the existing models is that most of studies in batch scheduling in the flow shop system used single flow shop system. In the single flow shop system, the orders or jobs that arrive from the customers are pooled in a single shipping/receiving unit. These orders are then transferred into the production facility if there is only one production facility or distributed into several production facilities, such as in a distributed flow shop or parallel flow shop. Due to production capacity constraint, in some studies, outsourcing is possible. Meanwhile, there is another environment where there are certain number of single flow shops that besides run autonomously, they also collaborate among each other. Each flow shop receives its own orders, but joint production is possible. Reciprocally, this flow shop can also handle orders from other flow shops to achieve global optimization.

There are several questions following this problem. The first question is what model that can be used as a basis for developing batch scheduling model for collaborative flow shop system. The second question is what optimization technique that is suitable for this model and how about its performance.

Based on these problem and question, this work aims to develop batch scheduling model for collaborative flow shops system. Its objective is to minimize make-span and total production cost. In this work, the flow shop produces multiple products, and each order also consists of multiple products. This model is developed by using non-dominated sorting algorithm (NSGA II) because this algorithm is widely proven in many multi objective optimization works, for example in workshop scheduling [21], automatic train operation [22], wireless sensor network [23], and so on.

The contributions of this work are as follows.

1) This work proposes a novel batch scheduling for collaborative flow shop system which is different from the common single flow shop system.

2) This work proposes a novel bilateral interchange between flow shops in the joint production to ensure take-and-give mechanism which is different from the centralized production system.

3) This model proposes a more profitable mechanism by implementing additional outsourcing charge so that both parties get benefit from the interchange mechanism.

This paper is organized as follows. The background, research purpose, contribution, and paper organization are described in section one. The latest literatures or works related to the batch scheduling is reviewed in section two. The proposed batch scheduling model is explained in section three. The simulation and result are explained in section four. The deeper analysis and findings are discussed and elaborated in section five. In the end, the work is concluded in section six.

II. Related Work

In this section, some latest works that focused on the batch scheduling problem, especially in flow shop system, were explored. This exploration focuses on the objective, method, and circumstance of each work. After the explanation, this section is closed by the summarization of these literatures and stating clear reasoning of why this work is needed. This exploration is as follows.

Wu et al. [3] proposed batch scheduling with storage constraints. The objective is minimizing make-span. The mixed integer linear programming (MILP) is used to formulate the model. In it, the manufacturing system produces multiple products. The process consists of two stages: production units and storage tank. Quality checking is conducted in the second stage. The scheduling is conducted in both stages.

Feng and Hu [8] proposed order batching and sequencing in a vegetable industry. The system was modeled as parallel machine with serial batching. There are two main processes: order picking and order packing. Both processes are conducted manually. In the beginning, similar orders are batched and scheduled. The objective was minimizing total completion time. They used genetic algorithm to solve the problem.

Ackerman, Fumero, and Montagna [2] developed batch scheduling in a multisite manufacturing environment. The system consists of multiple plants. Each plant produces goods in multiple stages. There are parallel non-identical processing units in every stage. In it, orders are sent by the customers with specific release date and due date. Meanwhile, the system must deliver complete orders before their due date. The objective was to minimize make-span over all installations. The decision parameters included number of batches, batch size, batch sequencing, and batch distribution. They used MILP to formulate the problem.

Chiu, Wu, Yeh, and Wang [20] developed batching model for production system that allows outsourcing mechanism. The outsourcing is allowed due to the limited capacity of the internal production facility. The system is a hybrid fabrication plant. Meanwhile, the outsourcing mechanism offers higher production cost. During the production process, there may be some non-conforming products. These non-conforming products then will be reworked. Its objective is to find optimum run time.

Hertrich, Weib, Ackerman, Heydrich, and Krumke [6] developed scheduling for the batching machines in the flow shop system. Its objective was to minimize make-span, total completion time, weighted total completion time, maximum lateness, total tardiness, and number of late jobs. In this system, there are certain number of installed machines. As a flow shop, the production system consists of several stages. Each step is handled by a single dedicated machine. The processing time are job-independent, or it is also known as proportionate flow shop. Each machine can handle multiple jobs at the same time in batch. This system adopts parallel batching machine so that the processing time of a batch remains fix. It is different from the serial batching machine where the processing time of a batch is the sum of the processing time of all jobs in a batch. In this work, the maximum batch size becomes constraint.

Maulidya, Suprayogi, Wangsaputra, and Halim [10] proposed batch scheduling model that is implemented for hybrid flow shop system. Its objective was to minimize total actual flow time. In this work, they used heuristic algorithm which consisted of two sub algorithms: the shortest processing time and variable neighborhood descent. In this flow shop system, the production process consists of three stages. The first stage consists of unrelated parallel machines for processing common and unique components. Then, these components are assembled in the second stage. Finally, these assembled products are then differentiated into various product types in the third stage.

This previous work was then improved by Suryandhini, Sukoyo, Suprayogi, and Halim [11]. Its objective was to minimize total actual flow time. They also used the heuristic algorithm. The improvement was including the sampling inspection in the three-stage flow shop system. In this flow shop system, the production process occurred in the first and second stages. Meanwhile, the inspection process was conducted in the third stage. The batch size and sequence are same in the production stages. In there, each batch consisted of one product type. Meanwhile, the third stage consisted of common inspection machine. The inspection mechanism was sampling and it followed Dodge-Romig sampling rule. During the production, there was non-conforming products, and they could not be reworked.

Ferretti and Zavanella [19] proposed batch scheduling model for general flow shop system. Its objective is to minimize total energy cost in the system. The circumstances were as follows. The flow shop consisted of two machines.

There was not any intermediate storage for interstage buffer. Batches size could be various with minimum and maximum sizes constraint. Processing time were constant, and it was independent on the batch size. The flow shop implemented no-wait scenario. The production capacity was limited. On the other side, demand had to be satisfied during the total time. The system was designed to produce identical parts. Processing time of the second stage was longer than the first stage.

Miao, Xia, Zhang, and Zou [5] proposed batch scheduling model with proportional deteriorating jobs. Its objective was to minimize total completion time. They used backward dynamic programming to solve the problem. The system adopted parallel batch machine so that the production facility could process several jobs simultaneously. The system also did not allow interruption. All jobs were ready at time zero. Batch could be full or not.

Mostafaei and Hanjunkoski [4] proposed formulation for batch scheduling. Its objective was to minimize the completion time. They used continuous time integer linear programming to formulate the model. The system allowed multiple intermediate due date. The production units were shared (not dedicated) and could produce multiple products.

Valdez-Navarro and Ricardez-Sandoval [17] integrated dynamic optimization and scheduling in the batch plant. The system was batch plant that consisted of multiple production units that could produces multiple products. Its objective was to maximize process revenue under fixed make-span. They used back-off decomposition algorithm to solve the problem. The demand was stochastic.

Ogun and Alabas-Uslu [18] proposed mathematical model to solve the batch scheduling problem in the system that produced multiple products. On the other side, each order consisted of multiple products too. Its objective was to minimize total tardiness and earliness. This system implemented parallel batch model. All parts of the same products that come from different orders could be processed in the same batch. The batch size was limited. This mathematical model was developed by using non-linear integer programming and linear integer programming.

Wu et al. [15] proposed a scheduling model that combined production and maintenance. Its objective was to minimize make-span. They used mixed integer linear programming to formulate the problem and model. This model was implemented in a batch plant that produced multiple products. In it, product degradation might occur. The system produced multiple grade products. It was a flow shop system. Each stage had several parallel units although the processing time was serial due to the quantity of products in a batch.

Yusriski, Astuti, Biksono, and Wardani [9] proposed integer batch scheduling model with single machine scenario. Its objective was to minimize total actual flow time. They compared three optimization models: longest due date (LDD), adjacent pairwise interchange (API), and permutation (PM) . In it, the jobs arrived with different due date. Every job consisted of one or more parts. A machine processed a job into number of batches. The decision parameters were jobs sequence, number of batches, batches sequence, and batches size. It followed serial batch model where the processing time of a batch was the sum of processing time of all parts in it.

Based on this exploration, these works can be summarized as follows. All of studies conducted single flow shop system with single or parallel production lines. Some works deployed parallel batching while others deployed serial batching. Most of them focused on minimizing make-span. The other objectives were to minimize TAFT and cost. Based on it, batch scheduling model that occurs in a collaborative flow shop system with multiple flow shops in the environment is still unexplored. So, developing a batch scheduling model in a collaborative flow shops environment is challenging.

III. Proposed Model

Before the proposed model is explained, first, the comparison between the environment in the single flow shop system and the collaborative flow shop system will be explained. The illustration of a single flow shop system is shown in Fig. 1.

The explanation of the Fig. 1 is as follows. There is a flow shop system with single door for shipping and receiving activities. The role of this door is to receive order from customers and deliver products to customers based on their order. This flow shop system has one plant or multiple parallel plants. The shipping/receiving unit then distributes orders to the plants. Then, after the plants complete the production, these products are then transferred back to the shipping/receiving unit to be delivered to the customers.

This condition is different from the collaborative flow shop system. In a collaborative system, there are multiple flow shops. Each flow shop interacts with its own customers. It receives orders from its customers and delivers the requested goods to them. Each flow shop has its own production facility. The production cost, production capacity, and processing time may be different among flow shop. In some circumstances, it is better to outsource some jobs or parts of jobs to other flow shops in the system, for example, due to the limited production capacity [20]. Reciprocally, a flow shop may get jobs or parts of jobs from other flow shops. This collaboration is illustrated in Fig. 2. In Fig. 2, there are two flow shops in the environment. Each flow shop has its own customers, shipping/receiving unit, and plant. Each flow shop only serves its own customers. Meanwhile, production outsourcing may occur between them.

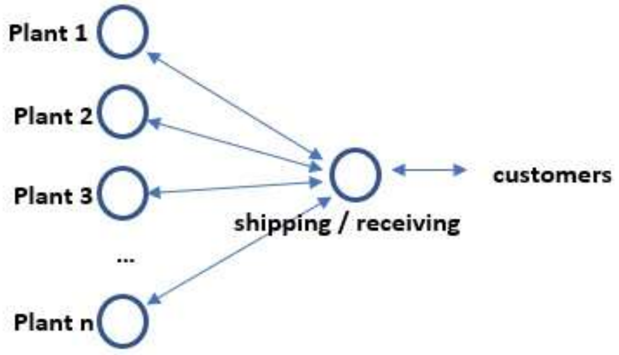

Fig. 1. Single Flow Shop System.

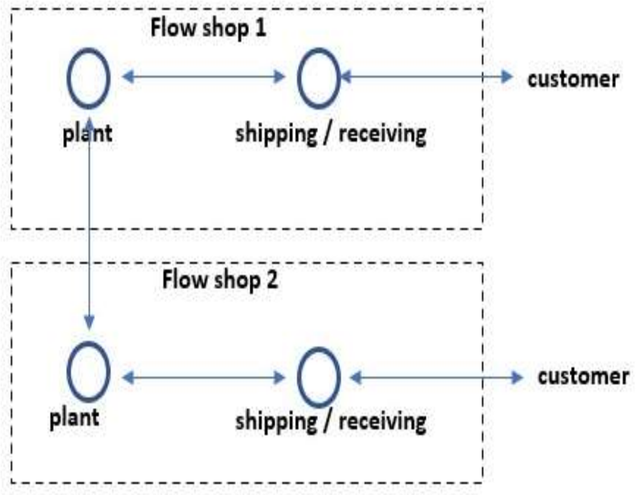

Fig. 2. Collaborative Flow Shop System.

In this work, the environment is a set of flow shops that interact to each other. Each flow shop has its own customers and production facility. The processing time, processing cost, and batch size may be different among flow shops. The objectives are to minimize make-span and total production cost. The assumptions in this model are as follows.

1) The flow shop is permutation so that once a sequence is generated, this sequence is fixed for all stages [13].

2) All flow shops have same production stages [24].

3) All products must pass all stages [24].

4) All products can be produced in any flow shop [25].

5) All orders are ready at time zero [26].

6) All production facilities are ready at time zero [26].

7) There is no due date.

8) Intermediate storage is unlimited.

9) Interruption and pre-emption are not allowed [26].

*10)*Parallel batching is implemented so that the batch processing time is independent to the number of jobs [7].

*11)*The batch size is determined based on the number of product units in a batch.

The process is described as follows. In the beginning, each flow shop receives orders from its customers. An order represents a job. A job consists of one or more products. A job is then split into packets based on the product. It means the number of packets in a job is equal to the number of products in a job. After all jobs in all flow shops are mapped into packets, then packet interchange among flow shop can be done. In this work, packet interchange is preferred to outsourcing because when a flow shop transfers a packet to be produced by other flow shop, it also receives packet from its counterpart. After packets interchange session ends, the next session is batching. Batching is conducted based on the product type and batch type. Each batch contains one or several packets with the same product.

Before further explanation of the model, here are notations that are used in this work.

a	action
b	batch
c	production cost
f	flow shop
f_{pr}	flow shop who processes the packet (packet processor)
f_{ow}	flow shop who owns the packet (packet owner)
f_{ms}	make-span fitness function
f_{tpc}	total production cost fitness function
g	product
o	order /job
p	packet
p_{sel}	selected packet
pos	packet position in a packet sequence
r	outsourcing rate
s	stage
q_b	batch's maximum quantity (batch size)
q_{cb}	batch's current quantity
q_p	packet's quantity
t_{end}	batch processing end time
t_p	processing time
t_{start}	batch processing start time
tr_{cr}	interchange threshold
tr_m	mutation threshold
B	set of batches
F	set of flow shops
G	set of products
O	set of orders /jobs
P	set of packets
S	set of stages

The first step is packets arrangement in every flow shop. The packets arrangement in the system can be modeled as $\{P(f_1), P(f_2), P(f_3), \ldots, P(f_{n(F)}),\}$. $P(f)$ is a set of packets that is owned by a flow shop and it can be modeled as $\{p(1,f), p(2,f), p(3,f), \ldots, p(n(P(f)),f)\}$. In other word, it is a two-dimensional list. The first dimension represents the flow shop. The second dimension represents the packets in a flow shop. The number of packets in every flow shop is formalized by using (1). The initial packet arrangement process is shown in algorithm 1.

$$n\big(P(f)\big) = \textstyle\sum_{\forall O(f)} n\big(G(o,f)\big) \tag{1}$$

algorithm 1: initial packets arrangement

```
for $i$ = 1 to $n(F)$ do
begin
    $j = 0$
    for $k$ = 1 to $n(O(f_i))$ do
    begin
        for $l$ = 1 to $n(P(o_k,f_i))$ do
        begin
            initialize($p_{j,i}$)
            $q_p(j,i) = q_g(k,i,l)$
            $j$++
        end for
    end for
end for
```

Algorithm 1 represents the packets arrangement based on the packet owner. Packet owner is a flow shop who handles the related order where the packet belongs to. In this collaborative work, packets are then arranged based on the packet processor. In the interchange process, a packet may be processed or produced by another flow shop. It means, the packet processor is a flow shop who executes or produces the packet. If the packet is not interchanged, then the packet owner is same as the packet processor. Otherwise, the packet owner is different from the packet processor.

The explanation of the parameters and variables in algorithm 1 is as follows. The initial packets arrangement is conducted in all flow shops which is represented by the looping from 1 to $n(F)$. Then, the second loop is conducted for all orders in the flow shop which is represented by $n(O(f_i))$. The third loop is conducted to arrange all packets in every order, which is represented by $n(P(o_k,f_i))$.

After all packets in all flow shops have been arranged, the next step is batch scheduling by using NSGA II. This model is the improvement of the previous multi-objective evolutionary algorithm (MOEA) by solving the difficulties of the MOEA: (1) computational complexities, (2) non-elitism, and (3) specified sharing parameters [27]. In general, the NSGA-II consists of six steps [28]: (1) population initialization, (2) non-dominated sorting, (3) crowd distance, (4) selection, (5) genetic operators, and (6) recombination. As a population-based method, in this work, a population or a solution represents a two-dimensional packet list in a system. Based on this general process [28], the algorithm of the collaborative batch scheduling in this work is as follows.

1) In the beginning, certain number of populations are set.

2) New off-springs are generated from the current population. The off-springs population is equal to the population size.

3) Cross over and mutation occurs. In this work, cross-over means interchanging the packets between two flow shops. Cross over process is formalized by using (2) to (8). Mutation means interchanging the position between two packets in the packet arrangement of a flow shop. This internal flow shop mutation occurs based on certain probability. This mutation is formalized by using (9) to (12).

4) Batching process occurs in all flow shops for all population.

5) Production process occurs based on the batch sequence.

6) Fitness calculation is conducted.

7) Non-dominated sorting is conducted based on the fitness functions (make-span and total production cost).

8) Crowd distance is conducted to sort solution in every group or front.

9) The first half of population is selected as the next parents and the rest are delisted from the population.

*10)*Step 2 and 9 is repeated until the maximum iteration is reached.

The initial packet interchange occurs in the initial population set up. Its process occurs based on stochastic process. This process is formalized by using (2) to (8). Equation (1) shows that interchange occurs only when the generated uniform random number is below the interchange threshold. Equation (3) shows that the interchange occurs by switching two selected packets in two selected flow shops. These flow shops are selected by using uniform random among flow shops in the system as it is shown in (4) and (5). Then, the packet position candidates are determined by using (6) and (7). Finally, the final packet position is the lowest position between these candidates to avoid misposition, due to the different number of packets in both flow shops, as it is shown in (8).

$$a = \begin{cases} interchange, U(0,1) < tr_{cr} \\ not\ interchange, else \end{cases}, \forall p\ in\ f \tag{2}$$

$$a = switch\big(p(f_{sel1}, pos_{sel}), p(f_{sel1}, pos_{sel})\big) \tag{3}$$

$$f_{sel1} = U\big(f_1, f_{n(F)}\big) \tag{4}$$

$$f_{sel2} = U\big(f_1, f_{n(F)}\big) \tag{5}$$

$$p_{sel1} = U\big(p_1, p_{n(P)}\big), \forall p\ in\ f_1 \tag{6}$$

$$p_{sel2} = U\big(p_1, p_{n(P)}\big), \forall p\ in\ f_2 \tag{7}$$

$$p_{sel} = min(p_{sel1}, p_{sel2}) \tag{8}$$

Crossover process is like the initial interchange process. Different from the initial interchange process, the crossover occurs in every iteration. The number of crossovers in every iteration is equal to the number of total flow shops in a system. The selection of the flow shops and packets follows (4) to (7).

Mutation occurs in all flow shops in every iteration. It occurs stochastically. This mechanism is formalized by using (9) to (12). Equation (9) shows that the mutation occurs when the generated uniform random number is less than the mutation threshold. The mutation occurs by switching the packets position of the two selected packets in a flow shop as it is shown in (10). Both packets are selected among packets in a flow shop as it is shown in (11) and (12).

$$a = \begin{cases} mutate, U(0,1) < tr_m \\ not\ mutate, else \end{cases}, \forall p\ in\ f \tag{9}$$

$$a = switch\big(p(f, pos_{sel1}), p(f, pos_{sel2})\big) \tag{10}$$

$$p_{sel1} = U\big(p_1, p_{n(P)}\big), \forall p\ in\ f \tag{11}$$

$$p_{sel2} = U\big(p_1, p_{n(P)}\big), \forall p\ in\ f \tag{12}$$

Hereafter, the batching sequencing is conducted by grouping packets based on the product. The product in a batch is homogeneous. Batching is also limited to the batch size. The process occurs sequentially from the first packet to the last packet in a flow shop. New batch will be created when there is not any available batch in a flow shop. Else, the packet is added into the available batch. This process is formalized by using (13). After allocating packet to batch, then the batch current quantity is accumulated.

$$a = \begin{cases} add, \exists b \wedge g_b = g_b \wedge \big(q_p + q_{cb}\big) \leq q_b \\ create, else \end{cases}, \forall p\ in\ f \tag{13}$$

The next step is production process. The batch scheduling occurs here. It follows permutation rule where batch overlapping is not permitted. The batch scheduling is formalized by using (14) to (16). Equation (14) and (15) is used to determine the batch start time. Meanwhile, the batch end time is determined by using (16).

$$t_{start}(b,s) = \begin{cases} 1, b = 1 \\ t_{end}(b-1,s)+1, b>1 \end{cases}, s = 1 \tag{14}$$

$$t_{start}(b,m) = \begin{cases} t_{end}(b,s-1)+1, b=1 \\ max\left(t_{end}(b-1,s), t_{end}(b,s-1)\right)+1, b>1 \end{cases}, s>1 \tag{15}$$

$$t_{end}(b,s) = t_{start}(b,s) + t_p(b,f) - 1 \tag{16}$$

After the production process ends, the next step is calculating the fitness function. Due to its multi objective model, there are two fitness functions: make-span and total production cost. These fitness functions calculation is formalized by using (17) to (19). Equation (17) shows that the make-span function is the maximum end time of the last stage of the last batch for all flow shops. Equation (18) shows that the total production cost is the accumulation of production cost of all packets of all flow shops. The packet production cost is calculated by using (19). In it, the packet production cost is obtained by multiplying the quantity of the packet and the unit production cost of the selected product, and its unit production cost is based on the packet processor. If the packet processor is different from the packet owner, then the outsourcing rate is included.

$$f_{ms} = max\left(t_{end}\left(f, n(B), n(S)\right)\right), \forall f \tag{17}$$

$$f_{tpc} = \sum_{\forall f} \sum_{\forall p} c(p) \tag{18}$$

$$c(p) = \begin{cases} q(p).c_u\left(f_{pr},g\right).(1+r), f_{pr} \neq f_{ow} \\ q(p).c_u\left(f_{pr},g\right), f_{pr} = f_{ow} \end{cases} \tag{19}$$

The next step is non-dominated sorting. In this work, the minimization approach is chosen. Based on the general rule in non-dominated sorting [27], in this work, solution A dominates solution B if these conditions meet.

1) Make-span of solution A is less than or equal to B.

2) Total production cost A is less than or equal to B.

3) Make-span of solution A is less than B or total production cost A is less than B.

Based on this requirement, every solution will be compared with all other solutions reciprocally. Solutions are then grouped based on the number of solutions they dominate. These groups are called as front. The groups are descending sorted. The last work is calculating the crowd distance. This process is used to rank the solutions inside the fronts. Because both fitness functions use different metric, the fitness of every solution is normalized first before calculated. In this work, the min-max normalization is used [29].

IV. Simulation

This proposed model is then implemented into simulation to observe and evaluate the performance of the model. In the simulation, this proposed model is compared with two non-collaborative models. The first non-collaborative model uses NSGA-II algorithm [30] and the second one uses adjacent pairwise interchange algorithm [9]. In this simulation, C-NSGA II represents the proposed model which is collaborative NSGA II; NC-NSGA II represents the non-collaborative NSGA II; and NC-API represents the non-collaborative adjacent pairwise interchange. The reasons of this comparison are as follows. The proposed model, as a collaborative model is compared with the non-collaborative model to observe its improvement as it is stated as the research purpose. The reason of comparing the NSGA II with the API is to compare the performance of the multi objective approach (NSGA II) and the single objective approach (API).

In this work, the observed parameters are make-span and total production cost. Meanwhile, the adjusted parameters are number of flow shops and number of customers. Besides these parameters, the value of other variables is shown in Table 1. In the beginning, some variables are generated randomly. The number of products, average batch size, average batch processing time, average number of products per order, and average product quantity are generated randomly and follow normal distribution.

The first simulation is simulating model with various number of flow shops. In this simulation, the number of flow shops ranges from 2 units to 10 units. In this simulation, the number of customers is set 50 units. Its objective is to observe the relationship between the number of flow shops and the observed parameters. The result is shown in Fig. 3 and Fig. 4.

TABLE I. Default Variables

Variables	Default Value
Number of stages	3 stages
Number of products	3 products
Outsourcing rate	10 percent
Average batch size	200 units
Average batch processing time	10 time-unit
Average number of products per order	3 products
Average per-product order quantity	30 units

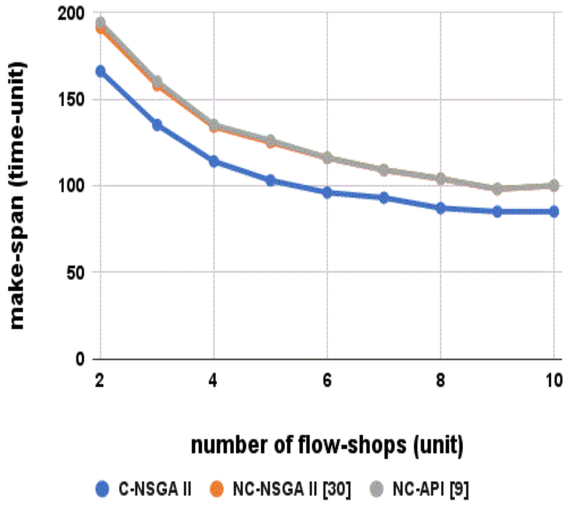

Fig. 3. Relation between the Number of Flow Shops and Make-Span.

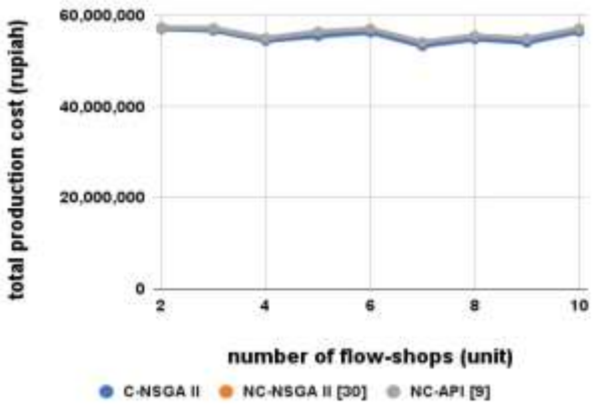

Fig. 4. Relation between the Number of Flow Shops and Total Production Cost.

Fig. 3 shows the relationship between the number of flow shops and make-span. It is shown that the make-span decreases due to the increasing of the number of flow shops. This decreasing of the make-span is the rational consequence of the increasing of the production units (flow shops) with the fixed number of customers. The make-span of the proposed model is the lowest one, compared with the existing NC-NSGA II [30] and NC-API [9]. It is 13 percent to 17 percent lower, with average 15 percent, than the existing models. Meanwhile, the make-span of the NC-NSGA II is almost equal to the NC-API with NC-NSGA II is slightly better.

Fig. 4 shows the relationship between the number of flow shops and total production cost. It is shown that the increasing of the number of flow shops does not affect the total production cost. The total production cost tends to fluctuate. By comparing among models, the total production cost is almost equal. The total production cost of the non-collaborative models is equal because the production is handled internally. Meanwhile, although almost equal, the total production cost of the proposed model is slightly lower than the existing non-collaborative models [9,30]. The total production cost of the proposed model is 0.6 percent to 1.8 percent lower, with average 1.3 percent, of the existing non-collaborative models.

The second simulation is simulating model with various number of customers. In this simulation, the number of customers ranges from 25 to 75 units. In this simulation, the number of flow shops is set 5 units. Its objective is to observe the relationship between the number of customers and the observed parameters. The result is shown in Fig. 5 and Fig. 6.

Fig. 5 shows that the make-span increases linearly due to the increasing of the number of customers. It is a rational consequence of the increasing jobs due to the fixed number of production units. The make-span of the existing non-collaborative models is almost equal with the NC-NSGA II [30] is slightly lower than the NC-API [9]. Meanwhile, the make-span of the proposed model is the lowest among models. The make-span of the proposed model is 10 to 17 percent lower, with average 14 percent, than the existing models.

Fig. 6 shows that the total production cost increases linearly due to the increasing of the number of customers. It occurs in all models. It is a rational consequence of the increasing jobs while on the other side, the number of production units is fixed. The total production cost is almost equal among all models. The total production cost of the NC-NSGA II [30] is equal to the NC-API because all jobs are handled internally. On the other side, the total production cost of the proposed model is slightly lower than the existing models. The total production cost of the proposed model is 0.3 to 3.5 percent lower, with average 1.6 percent, than the existing non-collaborative models. This gap is narrower when the number of customers is high.

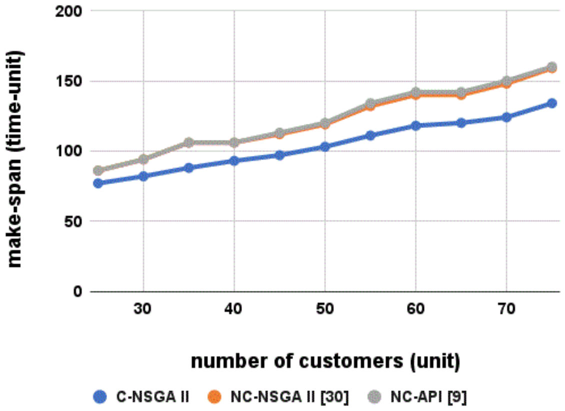

Fig. 5. Relation between the Number of Customers and Make-Span.

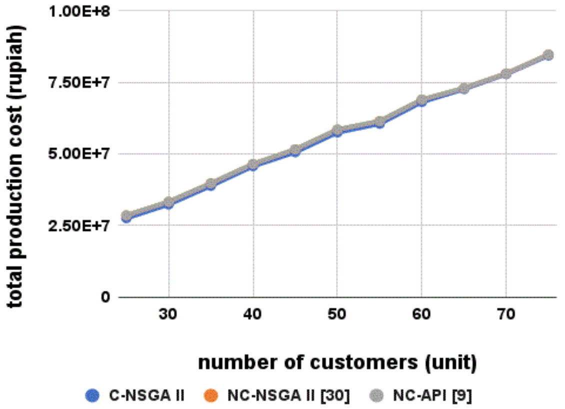

Fig. 6. Relation between the Number of Customers and Total Production Cost.

V. Discussion

Based on the simulation result, in general, the collaborative approach is better than the non-collaborative one. it is shown that the collaborative model outperforms the non-collaborative models in make-span and is slightly better in total production cost. The make-span of the proposed model is lower than the existing models, both the NC-NSGA II (30) and the NC-API [9], and its gap is wide. On the other side, although the gap is very narrow, the total production cost of the proposed model is still lower than the existing models.

In the make-span parameter, the make-span can be reduced in two ways. The first way is allocating jobs to the production unit which has more competitive processing time. The second way is by sequencing jobs into batches so that the number of batches can be minimized. By using collaborative approach, the proposed model tries to allocate jobs to the production unit (flow shop) which has lower batch processing time. This process is conducted during the iteration of the NSGA II. On the other side, in the existing non-collaborative models [9,30], optimization or make-span reduction is conducted by arranging jobs that must be processed internally so that the number of batches can be reduced. Meanwhile, this process also occurs in the non-collaborative model. It means, the collaborative model outperforms the non-collaborative models in minimizing the make-span because it runs two optimization processes. On the other side, the non-collaborative models [9,30] run only one optimization process. The result also shows that the non-collaborative NSGA II performs better than the non-collaborative API in the make-span aspect. It means that the NSGA II [30] can arranges jobs into batches better than the API [9] although the gap is very narrow.

In the total production cost parameter, the production cost can only be reduced by transferring packets to the production unit (flow shop) which has lower production cost. This process occurs in the collaborative proposed model only. Meanwhile, this process cannot be conducted in the non-collaborative models [9,30] which execute orders internally. Unfortunately, this advantage is reduced by the outsourcing cost which is charged to packets which are produced externally. This circumstance makes the total production cost gap between the collaborative model and the non-collaborative ones is not significant.

VI. Conclusion

This work shows that the proposed collaborative batch scheduling model meets the research objective in minimizing the make-span and the total production cost. The collaborative NSGA II model outperforms the non-collaborative models (NSGA II and adjacent pairwise interchange). Due to the simulation result, the make-span of the proposed model is lower than the non-collaborative models. The make-span of the proposed model is 10 to 17 percent lower than the existing non-collaborative models. The total production cost of the proposed model is 0.3 to 3.5 percent lower than the existing non-collaborative model.

There are several future research potentials that can be conducted based on this proposed collaborative model. This model can be expanded into collaborative parallel flow shops where every flow shop has several production lines. This model is also can be improved by adding several constraints, such as due date, no-wait scenario, penalty, etc.

References

[1] K. R. Baker and D. Trietsch, Principles of Sequencing and Scheduling, Wiley, Canada, 2009.

[2] S. Ackerman, Y. Fumero, and J. Montagna, “Incorporating batching decisions and operational constraints into the scheduling problem of multisite manufacturing environments”, International Journal of Industrial Engineering Computations, vol. 12, pp. 345-364, 2021.

[3] O. Wu, G. D. Ave, I. Harjunkoski, A. Bouaswaig, S. M. Schneider, M. Roth, and L. Imsland, “Short-term multiproduct batch scheduling considering storage features”, IFAC PapersOnLine, vol. 53, no. 2, pp. 11794-11799, 2020.

[4] H. Mostafaei and I. Harjunkoski, “Single reference grid continuous-time formulation for batch scheduling”, IFAC PapersOnLine, vol. 52, no. 1, pp. 832-837, 2019.

[5] C. Miao, Y. Xia, Y. Zhang, and J. Zou, “Batch scheduling with deteriorating jobs to minimize the total completion time”, Journal of Operations Research Society of China, vol. 1, pp. 377-383, 2013.

[6] C. Hertrich, C. Weib, H. Ackermann, S. Heydrich, and S. O. Krumke, “Scheduling a proportionate flow shop of batching machines”, Journal of Scheduling, vol. 23, pp. 575-593, 2020.

[7] J. W. Fowler and L. Monch, “A survey of scheduling with parallel batch (p-batch) processing”, European Journal of Operational Research, 2021, in press.

[8] X. Feng and X. Hu, “A heuristic solution approach to order batching and sequencing for manual picking and packing lines considering fatiguing effect”, Scientific Programming, article ID: 8863391, pp. 1-17, 2021.

[9] R. Yusriski, B. Astuti, D. Biksono, and T. A. Wardani, “A single machine multi-job integer batch scheduling problem with multi due date to minimize total actual flow time”, Decision Science Letters, vol. 10, pp. 231-240, 2021.

[10] R. Maulidya, Suprayogi, R. Wangsaputra, and A. H. Halim, “A batch scheduling model for a three-stage hybrid flowshop producing products with hierarchical assembly structures”, International Journal of Technology, vol. 11, no. 3, pp. 608-618, 2020.

[11] P. P. Suryandhini, Sukoyo, Suprayogi, and A. H. Halim, “A batch scheduling model for a three-stage flow shop with job and batch processors considering a sampling inspection to minimize expected total actual flow time”, Journal of Industrial Engineering and Management, vol. 14, no. 3, pp. 520-537, 2021.

[12] G. M. Komaki, S. Sheikh, and B. Malakooti, “Flow shop scheduling problems with assembly operations: a review and new trends”, International Journal of Production Research, pp. 1-30, 2018.

[13] K. Peng, L. Wen, R. Li, L. Gao, and X. Li, “An effective hybrid algorithm for permutation flow shop scheduling problem with setup time”, Procedia CIRP, vol. 72, pp. 1288-1292, 2018.

[14] D. A. Rossit, F. Tohme, M. Frutos, M. Safe, and O. C. Vasquez, “Critical paths of non-permutation and permutation flow shop scheduling problems”, International Journal of Industrial Engineering Computations, vol. 11, pp. 281-298, 2020.

[15] O. Wu, G. D. Ave, I. Harjunkoski, A. Bouaswaig, S. M. Schneider, M. Roth, and L. Imsland, “Optimal production and maintenance scheduling for a multiproduct batch plant considering degradation”, Computers and Chemical Engineering, vol. 135, article ID: 106734, pp. 1-14, 2020.

[16] F. Miquel, M. Frutos, F. Tohme, and D. A. Rossit, “A memetic algorithm for the integral OBP/OPP problem in a logistics distribution center”, Uncertain Supply Chain Management, vol. 7, pp. 203-214, 2019.

[17] Y. I. Valdez-Navarro and L. A. Ricardez-Sandoval, “Integration between dynamic optimization and scheduling of batch processes under uncertainty: a back-off approach”, IFAC PapersOnLine, vol. 52, no. 1, pp. 655-660, 2019.

[18] B. Ogun and L. Alabas-Uslu, “Mathematical models for a batch scheduling problem to minimize earliness and tardiness”, Journal of Industrial Engineering and Management, vol. 11, no. 3, pp. 390-405, 2018.

[19] I. Ferretti and L. E. Zavanella, “Batch energy scheduling problem with no-wait/blocking constraints for general flow-shop problem”, Procedia Manufacturing, vol. 42, pp. 273-280, 2020.

[20] S. W. Chiu, H. Y. Wu, T. M. Yeh, and Y. Wang, “Solving a hybrid batch production problem with unreliable equipment and quality reassurrance”, International Journal of Industrial Engineering Computations, vol. 12, pp. 235-248, 2021.

[21] W. Yahui, S. Ling, Z. Cai, F. Liuqiang, and J. Xiangjie, “NSGA-II algorithm and application for multi-objective flexible workshop scheduling”, Journal of Algorithms & Computational Technology, vol. 14, 2020.

[22] P. Chu, Y. Yu, D. Dong, H. Lin, and J. Yuan, “NSGA-II-based parameter tuning method and GM(1,1)-based development of fuzzy immune PID controller for automatic train operation system”,

Mathematical Problems in Engineering, article ID: 3731749, pp. 1-20, 2020.

[23] V. Kanwar and A. Kumar, "Multiobjective optimization-based DV-hop localization using NSGA-II algorithm for wireless sensor networks", International Journal of Communication Systems, vol. 33, no. 11, 2020.

[24] Y. Sun and X. Qi, "A DE-LS metaheuristic algorithm for hybrid flow-shop scheduling problem considering multiple requirements of customers", Scientific Programming, article ID: 8811391, pp. 1-14, 2020.

[25] B. Naderi and R. Ruiz, "The distributed permutation flowshop scheduling problem", Computers & Operations Research, vol. 37, no. 4, pp. 754-768, 2010.

[26] F. C. Cetinkaya, P. Yeloglu, and H. A. Catkamas, "Customer order scheduling with job-based processing on a single-machine to minimize the total completion time", International Journal of Industrial Engineering Computations, vol. 12, pp. 273-292, 2021.

[27] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II", IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182-197, 2002.

[28] Y. Yussoff, M. S. Ngadiman, and A. M. Zain, "Overview of NSGA-II for optimizing machining process parameters", Procedia Engineering, vol. 15, pp. 3978-3983, 2011.

[29] H. Benhar, A. Idri, and J. L. Fernandez-Aleman, "Data preprocessing for heart disease classification: a systematic literature review", Computer Methods and Programs in Biomedicine, vol. 195, article ID: 105635, 2020.

[30] N. Farmand, H. Zarei, and M. Rasti-Barzoki, "Two meta-heuristic algorithms for optimizing a multi-objective supply chain scheduling problem in an identical parallel machines environment", International Journal of Industrial Engineering Computations, vol. 12, pp. 249-272, 2021.