



## Collaborative Flow-shop Scheduling Using Simulated Annealing and First Price Sealed Bid Auction to Minimize Total Cost and Make-span

Purba Daru Kusuma<sup>1\*</sup>

Ratna Astuti Nugraheni<sup>1</sup>

<sup>1</sup>*Computer Engineering, Telkom University, Indonesia*

\* Corresponding author's Email: [purbodaru@telkomuniversity.ac.id](mailto:purbodaru@telkomuniversity.ac.id)

---

**Abstract:** This work proposes a new flow-shop scheduling model that consists of several flow-shops. Every flow-shop acts as an independent entity, but there is a collaboration among them. Although the relation between flow-shops and their customers is exclusive, collaboration through production sharing is possible. This circumstance is different from most studies in flow-shop scheduling problems (FSP), for example, parallel or distributed, where all jobs come from a single point and are then distributed to the production resources. In the multiple independent flow-shops, each flow-shop has its own processing time and production cost. Through collaboration, efficiency can be achieved in the make-span and total cost aspects, which becomes the objective of this work. This model is developed by combining the first price sealed bid auction and cloud theory-based simulated annealing. The first price sealed bid auction is conducted to minimize the total production cost. Meanwhile, the cloud theory-based simulated annealing is conducted to minimize the make-span. This model is then compared with the existing non-dominated sorting genetic algorithm (NSGA II) based flow-shop scheduling models. The first existing model is a parallel flow-shop, while the second one is a collaborative flow-shop. The simulation result shows that the proposed model outperforms the existing models in the total cost aspect. The proposed model creates a 13 to 29 percent lower total cost than the NSGA II-based parallel flow-shop. Meanwhile, the proposed model creates a 16 to 28 percent higher make-span than the NSGA II-based parallel flow-shop.

**Keywords:** Flow-shop scheduling, Multi-agent system, First price sealed bid, Collaborative system, Auction.

---

### 1. Introduction

Flow-shop scheduling (FSP) is a well-known production model in manufacturing systems besides job-shop, batch-shop, and so on. FSP is also a fascinating subject that many studies are conducted on due to the complexity of the production process. Besides, it also has a lot of variation and circumstances. Many studies in FSP were conducted based on specific circumstances, such as deteriorating jobs [1], sequent-dependent setup times [2], multiple requirements [3], waiting time [2], no-idle [4], blocking [5], and so on.

Besides the circumstance, many studies in FSP were conducted based on the specific metaheuristic optimization method. One most popular method is the evolutionary algorithm (EA) [6]. EA is a population-based optimization, and it is adopted from

the evolution process of living things. Genetic algorithm is a popular EA derivative method widely used in studies in FSP [1]. Besides, several other metaheuristic methods used in FSP studies are simulated annealing [7], tabu search [8], particle swarm optimization (PSO) [9], and so on.

Most of the studies in FSP were conducted by viewing the flow-shop as a single entity, no matter the flow-shop is a single flow-shop [10] or multiple flow-shops that can be parallel [5] or distributed flow shop [11]. All jobs come to a single point. Hereafter, these jobs will be distributed to the selected production resources in case that the system consists of multiple flow-shops. As a single entity, the assignment mechanism is authoritative.

Ironically, studies in FSP that consist of multiple flow-shops as independent entities were hard to find. Meanwhile, there are many production systems in the real world that represent multiple independent flow-

shops with collaboration among them, such as in the traditional crafts centrals. An example is hand-writing batik centrals in Solo, Indonesia. Hand-writing batik proceeds within several sequential steps. In every central, there are several hand-writing batik producers. Every producer has their jobs or orders. Meanwhile, a producer can also outsource its jobs to other producers in the central to minimize the processing time or production cost. However, every flow-shop has its independency to outsource its jobs or not. On the other side, a flow-shop is free to receive or decline the outsourced jobs.

Based on this problem, this work proposes an FSP model with multiple independent flow-shops in the system. The system is a multi-product production system, and a flow-shop can produce multiple products. A job is a single-product job. Meanwhile, the product of a job may be different from other jobs. Rather than centralized and authoritative, this model is collaborative and decentralized. The objective is to minimize total production cost and the maximum completion time (make-span). Minimizing the total production cost becomes the primary objective, while minimizing the make-span becomes the secondary objective.

This model uses first price sealed bid auction for the job assignment process and cloud theory-based simulated annealing for the scheduling process. The first price sealed bid auction is chosen due to its popularity in electronic auction [12], which one of its characteristics is anonymity.

The cloud theory-based simulated annealing is chosen due to several reasons. A simulated annealing method aims to achieve global optimization and avoid local optimal traps [6]. This objective is achieved by accepting a worse current solution in a certain degree of probability. It is different from other metaheuristic methods, such as genetic algorithms, in which new solutions are generated from current best solutions. Moreover, the cloud theory-based simulated annealing is chosen for its faster performance than the classic simulated annealing [13].

The contributions of this work are as follows.

- This work proposes a new collaborative flow-shop model that consists of multiple independent flow-shops that act autonomously.
- This work implements first price sealed bid auction as a job assignment process in the flow-shop system.

The remainder of this paper is organized as follows. The shortcoming studies in flow-shop scheduling problems are explored in section two. The

proposed model, which consists of system architecture and a mathematical model, is explained in section three. The simulation scenario and the result are shown in section four. The findings and deeper analysis due to the simulation result are discussed in section five. The conclusion of this work and future research potentials related to this work is explained in section six.

## 2. Related works

In general, a flow-shop is implemented in a mass production process where the jobs or products are proceeded within several steps where each job must pass all the steps in the same sequence [14]. Based on the sequence, there are two types of flow-shop: permutation FSP and non-permutation FSP. In the permutation FSP, once the sequence of the jobs is arranged, this sequence is fixed for all steps [14]. Meanwhile, in the non-permutation FSP, the sequence of the jobs may be different among steps [14].

Many studies in FSP were conducted with a view of FSP as a single entity. The FSP may consist of one flow-shop or multiple flow-shops. The multiple flow-shops can be parallel, distributed, or hybrid, and the single FSP can be implemented in a small or medium factory. Meanwhile, the flow-shop that consists of multiple production resources can be implemented as a large company with several production resources in a single location (parallel flow-shop) or many locations (distributed flow-shop). The circumstances of the shortcoming studies in FSP are listed in Table 1. These studies are sorted chronologically.

This exploration shows that many studies in FSP are conducted in several flow-shop types: multiple-machines, parallel, distributed, and hybrid. The theoretical explanation about these flow-shop models is as follows.

The two-machine flow-shop is a simple flow-shop system in a factory [14]. A factory has a certain number of jobs to be processed. On the other side, this factory has two machines. Each job must pass through these machines sequentially. For example, first, a job must be processed in the first machine. Hereafter, this job goes to the second machine for the next process.

The concept of a parallel flow-shop system is as follows [5]. A company has a factory that consists of several lines (flow-shops). These lines are identical, and each line consists of several machines. On the other side, this company has several jobs to be executed. A job can be distributed to any line, and each job must pass through all machines in line with the same order. The problem is how to distribute

Table 1. List of shortcomings FSP studies

Author	Flow-shop Type	No. of Companies	Assignment Model	Objectives
[15]	distributed flow-shop	single	authoritative	minimize make-span
[4]	multiple-machine flow-shop	single	authoritative	minimize cost
[16]	distributed flow-shop	single	authoritative	minimize make-span
[17]	distributed flow-shop	single	authoritative	minimize make-span
[18]	distributed flow-shop	single	authoritative	minimize make-span
[19]	single flow-shop	single	authoritative	minimize make-span
[20]	hybrid flow-shop	single	authoritative	minimize make-span minimize maximum tardiness minimize idle energy consumption
[11]	distributed flow-shop	single	authoritative	minimize make-span
[21]	two-machine flow-shop	single	authoritative	minimize make-span minimize total energy consumption
[22]	single flow-shop	single	authoritative	minimize make-span
[23]	single flow-shop	single	authoritative	minimize make-span
[24]	multiple-machine flow-shop	single	authoritative	minimize make-span
[25]	multiple-machine flow-shop	single	authoritative	minimize actual flow time
[8]	single machine flow-shop	single	authoritative	minimize total completion time
[5]	parallel flow-shop	single	authoritative	minimize make-span
[26]	single flow-shop	single	authoritative	minimize make-span
[10]	single flow-shop	single	authoritative	minimize total tardiness
[9]	parallel flow-shop	single	authoritative	minimize total cost minimize total time
this work	multiple flow-shops	multiple	voluntary	minimize production cost minimize make-span

these jobs into the lines and arrange each set of jobs in every line.

The concept of a distributed flow-shop system is as follows [18]. A company has multiple factories. On the other side, this company has several jobs or orders to be executed. Each factory is identical which consists of the same number of machines. These jobs are then distributed into the factories. All jobs will be processed in the same order. It means, wherever a job is allocated, it will be treated the same. The decision problem is in which factory each job will be sent to and in what kind of sequence will a set of jobs in every factory be arranged. The concept of distributed flow-shop is like the parallel flow-shop with different size and location of the factories.

The concept of a hybrid flow-shop is as follows [20]. A hybrid flow-shop is a derivative of the flow-shop system, which is more complicated. A factory has a production system where a job must be processed in several stages. There are several

machines at every stage, and the number of machines may be different among stages. A job must pass through all stages. Meanwhile, a job can be processed by any machine in a stage.

This explanation has shown that these existing studies could be implemented into a company's production system, whether its size is small, medium, or big. The assignment model is mandatory. It means every flow-shop must handle all jobs assigned to it. This approach is applied only in a company, whether it consists of a single or multiple flow-shops. In a system that consists of multiple companies, a company may accept or reject a job that is offered by other companies based on the possibility and profitability aspects. On the other side, in a multiple-company system, a company may or may not outsource its jobs. Ironically, studies in the flow-shop scheduling that consists of multiple companies (entities) with collaboration among them are still hard

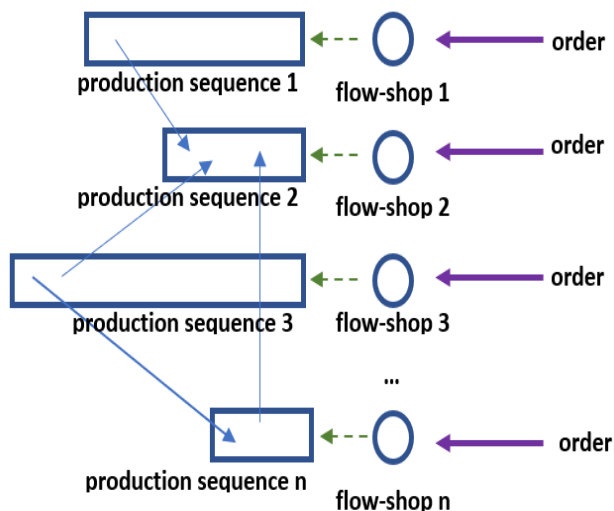


Figure. 1 System architecture

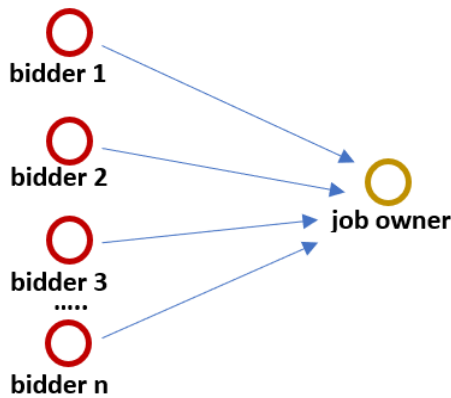


Figure. 2 Bidding mechanism

to find.

Based on this analysis, there is a potential to develop a new flow-shop scheduling model that consists of multiple companies or entities to become the scientific contribution of this work. In general, this model keeps the autonomy of every company in the system while promoting collaboration among them to achieve mutual efficiency.

The detailed explanation is as follows. Each company has its production resource and handles its customers independently. Meanwhile, due to the limited production resources, collaboration in the production process among them is possible but not mandatory or authoritative. It means a company may or may not outsource its jobs to other companies. On the other side, a company may accept or reject outsourcing offers from other companies. Moreover, the objectives of this proposed model are to minimize cost and make-span as they are (especially make-span) is the most common objective in the flow-shop scheduling studies.

### 3. Model

This model consists of several entities: flow-shops, jobs, products, steps, cost, and processing time. Flow-shops are entities that receive jobs from their customers and execute these jobs. Jobs are tasks that must be executed within several steps. Steps are stages that must be passed to execute a job. Cost is the production cost of a product. Processing time is the time needed to execute a job in a certain step. The system is a multi-product system. This flow-shop adopts the permutation approach. It means, once a sequence is arranged in a flow-shop, this sequence is applied for all steps.

In this system, every flow-shop receives orders or jobs from their customers. Then, every job must be executed by the job owner. Job owner is a flow-shop that receives the job from its customer. Meanwhile, a flow-shop can proceed a job cheaper than other flow-shops. Based on it, a job can be outsourced to another flow-shop. The flow-shop that proceeds the job is called a job processor. If the job proceeds internally, the job owner and the job processor are the same. Meanwhile, if the job is outsourced, then the job owner is different from the job processor. The number of jobs that are owned by every flow-shop is different among them. This mechanism is illustrated in Fig. 1.

The outsourcing mechanism is conducted by using an auction. Auction is chosen because it is a well-known mechanism for allocating goods, services, or objects to one or several parties. These parties will submit bids, which are usually sums of money [27]. In the auction system, there is a job that will be offered. Then, several bidders propose a bidding price to win the job. In this model, an outsourced (offered) job is owned by a job owner. The bidders are other flow-shops in the system. Every bidder offers a certain level of price, called bidding price, to the job owner. The flow-shop that wins the auction then becomes the job processor.

This model adopts first price sealed bid auction. As a first-price auction, the winner is the flow-shop who proposes the best (lowest) price, and the closed price is the first best price [27]. It is different from the Vickrey auction, where the winner is the bidder who proposes the best price, but the closed price is the second-best price [27]. As a sealed bid auction, a bidder can only observe its bidding price. It is different from the open bid price, like in Dutch auction or English auction, where the bidding price is open publicly [27]. This mechanism is illustrated in Fig. 2.

A bidder can propose a bidding price within its



Figure. 3 Bidding price range

reservation price and target point. A reservation price is the lowest possible proposed price, while a target point is the highest. In this model, the flow-shop’s production cost becomes the reservation price. A bidder will not propose a bidding price that is less than the production cost. Meanwhile, the flow-shop’s retail price becomes the target point. The retail price is the price offered by the flow-shop to its customer for the same product. It means the inter-flow-shop price will not surpass its retail price. Fig. 3 illustrates the bidding price range.

There are several assumptions applied in this work. These assumptions are as follows.

- A job is proceeded by only one flow-shop.
- A job consists of one product.
- A job can be proceeded by any flow-shop.
- All jobs are ready at time zero.
- A flow-shop can proceed all products.
- The retail price of a product must be higher than its production cost in a flow-shop.
- Once a production sequence is arranged, this sequence will not change during the processing period.
- The processing time of every job is known in advance.
- The setup time is zero.
- All machines are ready at time zero.

There are several annotations used in the mathematical model, as follows.

The model consists of two steps. The first step is assigning jobs to their job processor. The second step is the flow shop scheduling, which occurs in every job processor. The objective of the first step is to minimize the total cost (primary objective). The objective of the second step is to minimize the make-span (secondary objective). The first step is conducted by using the first price sealed bid auction. The second step is conducted by using cloud theory-based simulated annealing. (CSA).

The job assignment process consists of three steps, starting with job selection. The second step is bidding, and the third step is winner determination. The algorithm of this job assignment process is shown in algorithm 1.

$c$	cost
$c_{tot}$	total cost
$f$	flow-shop
$f_{ab}$	available flow-shop
$F_{ab}$	set of available flow-shops
$f_{pr}$	job’s processor
$f_{ow}$	job’s owner
$f_{se}$	selected flow-shop
$f_{wi}$	auction winner
$F$	set of flow-shops
$F_{bp}$	set of flow-shops that send bidding price (bidding pool)
$g$	good/product
$G$	set of products
$h$	fitness function
$j$	job
$j_{la}$	last job in a jobs sequence
$j_{pr}$	processed job
$J_{pr}$	set of processed jobs
$j_{se}$	selected job
$J$	set of jobs
$k$	coefficient
$m_r$	retail margin
$n_{bj}$	number of bid jobs
$n_{ow}$	number of owned jobs
$n_{pr}$	number of processed jobs
$n_{mp}$	maximum number of processed jobs
$p$	Price
$p_{bid}$	bidding price
$p_{re}$	reservation price
$p_{sel}$	selected price
$p_{ta}$	target point price
$r_o$	outsourcing threshold
$q$	queue
$s$	step
$s_{la}$	last step of a job
$S$	set of steps
$sol_{cur}$	current solution
$sol_{bes}$	best solution
$sol_{fin}$	final solution
$t$	time
$t_{en}$	end-time
$t_{ma}$	maximum processing time (make-span)
$T$	temperature
$T_{in}$	initial temperature
$T_{en}$	end temperature
$\Delta h$	fitness gap
$\Delta T$	temperature degradation rate

The explanation of algorithm 1 is as follows. The bidding process is conducted to the certain number of jobs ( $n_{bj}$ ) that is determined previously. There are

three processes in every iteration: job selection,

```

algorithm 1: jobs assignment
1  i = 0
2  while i < nbj do
3  begin
4  jse = U(J)
5  if U(0,1) < ro(fow(jse)) then
6  Fbp = bidding (F, jse)
7  fpr(jse) = select (Fbp, fow(jse), jse)
8  end if
9  i = i + 1
10 end while
    
```

bidding, and winner selection. The job is selected stochastically, and it follows a uniform distribution. It is shown that a job can be bid several times, while other jobs may not have the opportunity to be bid. Hereafter, the bidding process is conducted by offering the opportunity to the flow-shops in the system to propose a bidding price for this selected job. These proposals are then collected into a bidding pool. The bidding mechanism is conducted in a single round, which means a flow-shop can only send a proposal once, and this proposal cannot be revised or retracted.

Meanwhile, a flow-shop can send other proposals for other bid jobs. Finally, the winner is determined based on the proposals in the bidding pool. The detailed mechanism is formalized by using Eq. (1) to Eq. (7).

$$f_{ab}(j_{se}) = f \in F | f \neq f_{ow}(j_{se}) \wedge n_{pr}(f) < n_{mp} \tag{1}$$

$$p_{bi}(f, j) = U(p_{re}(f, j), p_{ta}(f, j)) \tag{2}$$

$$p_{re}(f, j) = c(f, p(j)) \tag{3}$$

$$p_{ta}(f, j) = (1 + m_r(f)) \cdot p_{re}(f, j) \tag{4}$$

$$F_{bp}(j) = \{f | f \in F_{ab}(j)\} \tag{5}$$

$$f_{wi}(j) = f | f \in F_{bp}(j) \wedge \min(p_{bi}(f, j)) \tag{6}$$

$$f_{pr}(j) = \begin{cases} f_{wi}(j), p_{bi}(f_{wi}(j)) < c(f_{ow}(j)) \\ f_{ow}(j), else \end{cases} \tag{7}$$

The explanation of Eq. (1) to Eq. (7) is as follows. Eq. (1) shows that a flow-shop can send a bidding proposal for a job if it is not the job-owner, and its processed jobs are less than the maximum number of processed jobs. Eq. (2) shows that the bidding price

is determined stochastically, and it follows uniform distribution from the reservation price and target point. Eq. (3) shows that the reservation price is the production cost of the flow-shop to produce the job-related product. Eq. (4) shows that the target point is the cost pluses retail margin. Eq. (5) shows that the bidding pool consists of all available flow-shops for the selected job. Eq. (6) shows that the auction winner is a flow-shop in the bidding pool with the lowest bidding price. Finally, Eq. (7) determines the job processor, whether the job owner of the auction winner. The auction winner becomes the job processor if its bidding price is less than the job owner’s cost.

The second step is job sequencing within the flow-shop. This work is conducted by using cloud-theory-based simulated annealing. This step consists of two rounds: initiation and iteration. The iteration is conducted after the initiation. As a population-based metaheuristic method, this model consists of several independent solutions. Both initiation and iteration are conducted in every solution. The best solution is the chosen among solutions whose fitness value is the lowest one. As mentioned before, the objective of this step is to minimize the maximum completion time, i.e., make-span. This objective is formalized by using Eq. (8) to Eq. (10).

$$h = \min(t_{ma}(F)) \tag{8}$$

$$t_{ma}(F) = \max(t_{ma}(f)), \forall f \in F \tag{9}$$

$$t_{ma}(f) = t_{en}(s_{la}(j)), j = j_{la}(J_{pr}(f)) \tag{10}$$

The explanation of Eq. (8) to Eq. (10) is as follows. Eq. (8) shows that the objective is to minimize the maximum completion time of all flow-shops. Eq. (9) shows that the maximum completion of all flow-shops is obtained from the flow-shop, whose maximum completion time is the highest. Eq. (10) shows that the maximum completion time of a flow-shop is obtained from the end time of the last step of the last job that is proceeded by the flow-shop.

The jobs sequence is scrambled in the initiation process. The objective is to differentiate the job sequence in the same flow-shop among populations. This process is conducted by using the pairwise interchange method. This scrambling process is shown in algorithm 2. Both interchanged jobs in the sequence ( $j(q_1)$  and  $j(q_2)$ ) is selected randomly among the jobs, and it follows a uniform distribution. The scrambling process occurs for all jobs. The process in algorithm 2 is conducted for all solutions.

The next round is the iteration. The iteration process is conducted for all populations. As a

```

algorithm 2: scrambling mechanism
1  i = 0
2  while i < n(J) do
3  begin
4  q1 = U(n(J))
5  q2 = U(n(J))
6  buf = j(q1)
7  j(q1) = j(q2)
8  j(q2) = buf
9  end while
    
```

```

algorithm 3: iteration process
1  T = Tin
2  while T > Ten do
3  begin
4  i = 0
5  while i < nima do
6  pairwise-interchange ()
7  Δh = h(solcur) - h(solbes)
8  if Δh < 0 then
9  solbes = solcur
10 else
11 if U(0,1) < exp(Δh/(k.T)) then
12 solbes = solcur
13 end if
14 i = i + 1
15 end while
16 T = T - ΔT
17 end while
    
```

simulated annealing method, the iteration starts from the initial high temperature and ends when the targeted low temperature is reached. There is a certain number of iterations in every temperature level. Every iteration consists of several processes: pairwise interchange, fitness recalculation, best solution updating. This iteration process is formalized by using algorithm 3. The final solution is determined by using Eq. (11). Eq. (11) shows that the final solution is obtained from the best solution among a population with the best fitness value.

$$sol_{fin} = sol_{bes} \in Sol_{bes} \wedge \min(h(sol_{bes})) \quad (11)$$

The pairwise interchange mechanism is conducted by interchanging jobs position within the job sequence of a flow-shop. There is only one flow-shop that runs a pairwise interchange process in every iteration. The pairwise interchange process in every iteration is formalized using Eq. (12) and Eq. (13).

Table 2. Default parameters

Parameters	Value
$n(G)$	5 units
$n(F)$	5 units
$n(S)$	4 units
$t_{av}$	5 time-unit
$c_{av}$	20 cost-unit

$$f_{se} = U(F) \quad (12)$$

$$j_{se} = U(J_{pr}(f_{se})) \quad (13)$$

The explanation of Eq. (12) and Eq. (13) is as follows. Eq. (12) shows that the flow-shop is selected randomly among flow-shops in the system, following a uniform distribution. Eq. (13) shows that the job is also selected randomly among jobs that are proceeded by the flow-shop, and it also follows a uniform distribution. The mechanism in Eq. (13) is conducted for both jobs that will be interchanged.

#### 4. Simulation and result

This proposed model is then implemented into flow-shop scheduling simulation to observe the performance of the model. The observed parameters are the make-span and the total cost, as they are stated as objectives of this work. Moreover, the make-span and cost are commonly observed parameters used in many studies in the FSP as shown in Table 1. This simulation is conducted to observe the relation between the number of jobs to the observed parameters. The number of jobs ranges from 20 to 100 units. Besides, there are several default parameters and their value. These default parameters are listed in Table 2. The processing time for every step and the production cost for every product in every flow-shop are generated randomly and follow a uniform distribution.

This proposed model is compared with two shortcomings of FSP models. Both compared models used NSGA II. The first model is a multi-objective parallel flow-shop scheduling model [9]. The second model is a collaborative flow-shop scheduling model [28].

The reason of choosing these two models is as follows. The first model represents the parallel flow-shop scheduling model so that the scheduling processes are centralized and authoritative. All jobs can proceed in any flow-shop in the system without the job processor can reject [5]. The second model represents the collaborative flow-shop scheduling



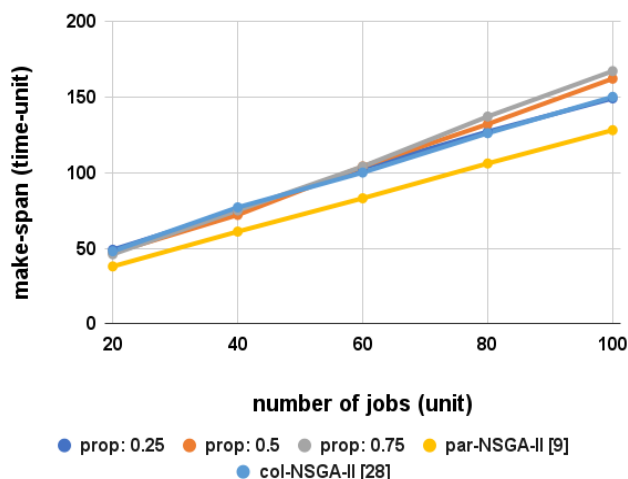


Figure. 4 Relation between number of jobs and make-span

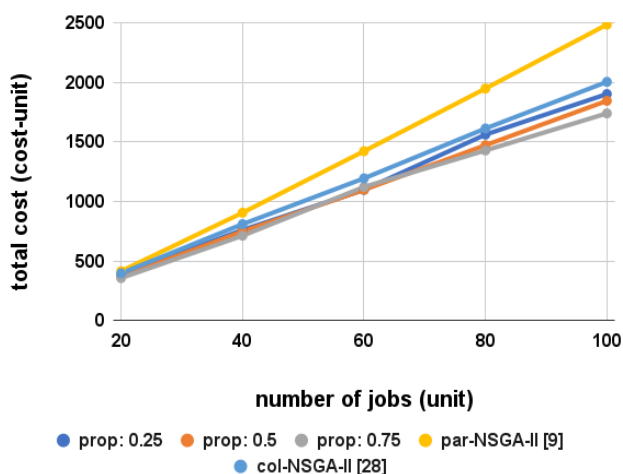


Figure. 5 Relation between number of jobs and total cost

model. The collaboration is conducted based on an exchange mechanism. It means a flow-shop can outsource a job to another flow-shop only if it also proceeds a job from its opponent. The outsourcing mechanism is still mandatory. The similarity between these two existing models is that a job can be processed in any flow-shop in the system. The difference is that in the first model [9], the job can be distributed easily while in the second model [28], the job can be outsourced only with reciprocal exchange.

The outsourcing price of the existing models is in the middle between the retail price and the production cost. The proposed model is conducted in three scenarios due to the outsourcing threshold: 0.25; 0.5; and 0.75. The result is shown in Fig. 4 and Fig. 5.

Fig. 4 shows that the make-span is proportional to the number of jobs occurring for all models. Compared among models, the NSGA II-based

parallel flow shop model [9] creates the lowest make-span. Meanwhile, the proposed model with a 0.75 outsourcing threshold creates the highest make-span. Comparing among three threshold values, the model with a 0.25 outsourcing threshold creates the lowest make-span. It means that the increase of the outsourcing threshold makes the make-span increase. The make-span of the proposed model with a 0.25 outsourcing threshold is almost equal to the NSGA II-based collaborative model [28]. The proposed model with a 0.25 threshold creates a 28 percent higher make-span than the NSGA II-based parallel flow-shop [9] when the number of jobs is low (20 units). This value decreases to 16 percent when the number of jobs is high (100 units).

Fig. 5 shows that the total cost is proportional to the number of jobs. It occurs for all models. Compared among models, the proposed model creates the lowest total cost. It occurs for all outsourcing thresholds (0.25, 0.5, and 0.75). The proposed model with a 0.75 outsourcing threshold performs as the best model.

Meanwhile, the NSGA II-based parallel flow-shop model performs as the worst model, and the NSGA II-based collaborative model performs the second worst one. Fig. 5 also shows that a higher outsourcing threshold decreases the total cost, especially for a high number of jobs. The proposed model with a 0.75 outsourcing threshold creates a 13 percent lower total cost than the NSGA II-based parallel flow-shop model [9] when the number of jobs is low (20 units). This value increases to 29 percent when the number of jobs is high (100 units).

## 5. Discussion

Several findings are obtained from the simulation result. First, the proposed collaborative model is proven to minimize total cost. It is achieved in two ways. The first way is conducting the auction. This first-price sealed-bid auction allocates the jobs to the flow-shop that offers the lowest bidding price. The lowest bidding price is proposed by combining flow-shop with low production cost or low outsourcing margin. The second way is conducting job assignments first before the scheduling process. It means the proposed model prioritizes the cost rather than the make-span.

The second finding is that the proposed model creates the highest make-span because of its limited optimization space. The proposed model can only minimize the make-span by interchanging jobs within the flow-shop. On the other side, the existing models, both the parallel flow-shop [9] and the collaborative flow-shop [28], conducts inter flow-



shop jobs interchange besides the intra flow-shop jobs interchange. It is based on the concept of the parallel flow-shop system where a job can be distributed to any lines in the system [5].

Third, the parallel flow-shop's second aspect that can create the lowest make-span is the distribution of the balanced jobs among flow-shops [9]. This circumstance is not found in the collaborative models, both the NSGA II-based model [28] or the proposed model. The collaborative model does not concern with balancing the workload among flow-shops or production resources. The outsourcing mechanism in the collaborative NSGA II flow-shop model guarantees the number of jobs proceeded by every flow-shop remains fixed [28]. It is because a flow-shop can outsource its in-hand job only if it also receives its opponent's in-hand job. In the proposed model, every flow-shop's number of jobs proceeded remains dynamic due to the auction process. Although the maximum number of proceeded flow-shop limits the number of jobs that can be proceeded, this mechanism does not guarantee that the load among flow-shops is balanced or equal.

## 6. Conclusion

This work has demonstrated that the collaborative flow-shop model can become the alternative to the existing flow-shop models (single, parallel, or distributed one). Moreover, this work also shows that the decentralized assignment model can be used as an alternative rather than the common centralized and authoritative assignment model. The first price sealed bid auction is proven in assigning jobs to the lowest cost production resource, although it sacrifices the make-span. The proposed model creates 13 to 29 percent lower total cost than the existing models while sacrificing 16 to 28 percent higher make-span. This collaborative flow-shop model has proven to become an alternative for the existing centralized flow-shop models. It democratizes the interaction among production resources rather than using an authoritative mechanism. Meanwhile, making the production system becomes more efficient is still challenging. In the future, this collaborative model can be expanded to various and more complex circumstances in the production system.

## Conflicts of interest

The authors declare no conflict of interest.

## Author contributions

Conceptualization: Kusuma; methodology: Kusuma, software: Kusuma, formal analysis:

Kusuma and Nugrahaeni; writing-original paper draft: Kusuma; writing-review and editing: Nugrahaeni.

## Acknowledgments

This work is funded and supported by Telkom University, Indonesia.

## References

- [1] H. Xuan, H. Zhang, and B. Li, "An Improved Discrete Artificial Bee Colony Algorithm for Flexible Flowshop Scheduling with Step Deteriorating Jobs and Sequence-dependent Setup Times", *Mathematical Problems in Engineering*, article ID: 8520503, pp. 1-13, 2019.
- [2] J. Y. Lee, "A Genetic Algorithm for a Two-machine Flowshop with a Limited Waiting Time Constraint and Sequence-dependent Setup Times", *Mathematical Problems in Engineering*, article ID: 8833645, pp. 1-13, 2020.
- [3] 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.
- [4] D. Gupta and H. Singh, "A Heuristic Approach to  $n \times m$  Flow Shop Scheduling Problem in which Processing Times are Associated with Their Respective Probabilities with No-idle Constraint", *ISRN Operations Research*, article ID: 948541, pp. 1-9, 2013.
- [5] I. Ribas and R. Companys, "A Computational Evaluation of Constructive Heuristics for the Parallel Blocking Flow Shop Problem with Sequence-dependent Setup Times", *International Journal of Industrial Engineering Computations*, Vol. 12, pp. 321–328, 2021.
- [6] A. P. Engelbrecht, *Computational Intelligence: an Introduction*, John Wiley & Sons, 2007.
- [7] C. L. Hsu, W. C. Lin, L. Duan, J. R. Liao, C. C. Wu, and J. H. Chen, "A Robust Two-machine Flow-shop Scheduling Model with Scenario-dependent Processing Times", *Discrete Dynamics in Nature and Society*, article ID: 3530701, pp. 1-16, 2020.
- [8] F. Cetinkaya, P. Yeloglu, and H. Catmakas, "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.
- [9] N. Farmand, H. Zarei, and M. R. 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.
- [10] T. T. Yamada, M. S. Nagano, and H. H. Miyata, “Minimization of Total Tardiness in no-wait Flowshop Production Systems with Preventive Maintenance”, *International Journal of Industrial Engineering Computations*, Vol. 12, pp. 415-426, 2021.
- [11] R. Ruiz, Q. K. Pan, and B. Naderi, “Iterated Greedy Methods for the Distributed Permutation Flowshop Scheduling Problem”, *Omega*, Vol. 83, pp. 213-222, 2019.
- [12] Z. Guo, Y. Fu, and C. Cao, “Secure First-Price Sealed-Bid Auction Scheme”, *EURASIP Journal on Information Security*, Vol. 2017, article No. 16, 2017.
- [13] E. Torabzadeh and M. Zandieh, “Cloud Theory-based Simulated Annealing Approach for Scheduling in the Two-stage Assembly Flowshop”, *Advances in Engineering Software*, Vol. 41, No. 10-11, pp. 1238-1243, 2010.
- [14] K. R. Baker and D. Trietsch, *Principle of Sequencing and Scheduling*, John Wiley & Sons, New Jersey, 2009.
- [15] B. Naderi and R. Ruiz, “The Distributed Permutation Flowshop Scheduling Problem”, *Computers & Operations Research*, Vol. 37, No. 4, pp. 754-768, 2010.
- [16] B. Naderi and R. Ruiz, “A Scatter Search Algorithm for the Distributed Permutation Flowshop Scheduling Problem”, *European Journal of Operational Research*, Vol. 239, No. 2, pp. 323-334, 2014.
- [17] Y. Xu, L. Wang, S. Wang, and M. Liu, “An Effective Hybrid Immune Algorithm for Solving the Distributed Permutation Flow-Shop Scheduling Problem”, *Engineering Optimization*, Vol. 46, No. 9, pp. 1269-1283, 2014.
- [18] V. F. Viagas and J. M. Framinan, “A Bounded-search Iterated Greedy Algorithm for the Distributed Permutation Flowshop Scheduling Problem”, *International Journal of Production Research*, Vol. 53, No. 4, pp. 1111-1123, 2015.
- [19] 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.
- [20] K. Geng, C. Ye, L. Cao, and L. Liu, “Multi-Objective Reentrant Hybrid Flowshop Scheduling with Machines Turning on and off Control Strategy Using Improved Multi-Verse Optimizer Algorithm”, *Mathematical Problems in Engineering*, article ID: 2573873, pp. 1-19, 2019.
- [21] S. Assia, I. E. Abbassi, A. E. Barkany, M. Darcherif, and A. E. Biyaali, “Green Scheduling of Jobs and Flexible Periods of Maintenance in a Two-Machine Flowshop to Minimize Makespan, a Measure of Service Level and Total Energy Consumption”, *Advances in Operations Research*, article ID: 9732563, pp. 1-9, 2020.
- [22] D. A. Rossit, F. Tohme, M. Frutos, M. Safe, and O. C. Vasquez, “Critical Paths on non-Permutation and Permutation Flow Shop Scheduling Problems”, *International Journal of Industrial Engineering Computations*, Vol. 11, pp. 281-298, 2020.
- [23] A. K. Mishra and D. Shrivastava, “A Discrete Jaya Algorithm for Permutation Flow-shop Scheduling Problem”, *International Journal of Industrial Engineering Computations*, Vol. 11, pp. 415-428, 2020.
- [24] M. I. Takano and M. S. Nagano, “Solving the Permutation Flow Shop Problem with Blocking and Setup Time Constraints”, *International Journal of Industrial Engineering Computations*, Vol. 11, pp. 469-480, 2020.
- [25] D. Kurniawan, A. C. Raja, Suprayogi, and A. H. Halim, “A Flow Shop Batch Scheduling and Operator Assignment Model with Time-Changing Effects of Learning and Forgetting to Minimize Total Actual Flow Time”, *Journal of Industrial Engineering and Management*, Vol. 13, No. 3, pp. 546-564, 2020.
- [26] M. A. Basset, R. Mohamed, M. Abouhawwash, R. K. Chakraborty, and M. J. Ryan, “A Simple and Effective Approach for Tackling the Permutation Flow Shop Scheduling Problem”, *Mathematics*, Vol. 9, article ID: 270, pp. 1-23, 2021.
- [27] L. Arozamena, A. Fioriti, and F. Weinschelbaum, “From Auction Theory to Market Design: Paul Milgrom and Robert Wilson’s Contributions to Economics”, *Estudios Economicos*, Vol. 38, No. 76, pp. 279-296, 2021.
- [28] P. D. Kusuma, “Multi-objective Batch Scheduling in Collaborative Multi-product Flow Shop System by Using Non-dominated Sorting Genetic Algorithm”, *International Journal of Advanced Computer Science and Application*, Vol. 12, No. 9, pp. 349-357, 2021.