Population-Based Metaheuristic Algorithms for a Hybrid Batch-Continuous Production Scheduling Problem in a Distributed Pharmaceutical Supply Chain

Seung Jae Lee and Byung Soo Kim *

Posted Date: 13 March 2026

doi: 10.20944/preprints202603.1065.v1

Keywords: scheduling; pharmaceutical supply chain; batch and continuous flowshop-line; order split; genetic algorithm; particle swarm optimization

 doi:10.20944/preprints202603.1065.v1

Article

Population-Based Metaheuristic Algorithms for a Hybrid Batch-Continuous Production Scheduling Problem in a Distributed Pharmaceutical Supply Chain

Seung Jae Lee and Byung Soo Kim *

Department of Industrial and Management Engineering, Incheon National University, 119, Academy-ro, Yeonsu-gu, Incheon, 22012, Republic of Korea

* Correspondence: bskim@inu.ac.kr; Tel: +82-32-835-8482; Fax: +82-32-835-0777

Abstract

We study a pharmaceutical scheduling problem with hybrid batch-continuous manufacturing process in a distributed supply chain. The supply chain consists of heterogeneous plants and one distribution center. Each plant adopts an unrelated permutation flow shop layout consisting of a hybrid batch-continuous production line. Each pharmaceutical order is split and produced in multi-production sites located in various regions. The pharmaceutical medicines manufactured by the production sites are directly shipped to a distribution center To minimize the makespan, we formulate the addressed scheduling problem as a mathematical model. To solve this model, we propose four metaheuristics variants by applying two population-based metaheuristics to two distinct solution structures. We compare the proposed metaheuristics to evaluate their performance in the numerical experiments. Additionally, we present managerial insights through sensitivity analysis.

Keywords: scheduling; pharmaceutical supply chain; batch and continuous flowshop-line; order split; genetic algorithm; particle swarm optimization

1. Introduction

Pharmaceutical medicines for treating or preventing diseases are divided into two main categories depending upon their manufacturing requirements: small molecules chemically synthesized and biologics derived from biological sources [1]. The demand on the pharmaceutical medicines is continuously increasing due to the impact of the COVID-19 pandemic, growing healthcare, etc. By 2027, spending on global medicine is predicted to reach $1.9 trillion [2]. One of the key concerns of the pharmaceutical manufacturing industry is establishing efficient production plans and scheduling to quickly respond to the demand fluctuations during pandemic periods [1]. To agilely handle the supply chain risk caused by the demand fluctuation,

In pharmaceutical supply chain, the production companies have been shifted their manufacturing plants from centralized to decentralized. Due to the globalization of markets, technological advancements, and other various factors in the pharmaceutical supply chain, it is common to operate multi-production sites in various regions effectively, and integrating these operations is very important [3–5]. In a distributed manufacturing (DM) strategy, production orders are decentralized from one large plant to multiple smaller sites to enhance responsiveness and system reliability while reducing production costs and shutdown risks. The COVID-19 pandemic has accelerated the shift toward decentralized pharmaceutical manufacturing systems [6,7]. According to Marques et al [8], DM strategy is an effective way for developing countries, where epidemics occur frequently, to provide flexibility in response to sudden demand for medicines.

 doi:10.20944/preprints202603.1065.v1

In pharmaceutical manufacturing process, a hybrid batch-continuous manufacturing process has recently gained attention to pharmaceutical manufacturing plants as an alternative manufacturing process that enables flexible responses to uncertain pharmaceutical demand [8–12]. Pharmaceutical batch manufacturing simultaneously feeds inputs and produces intermediates or final products at once. In contrast, continuous manufacturing steadily inputs materials and continuously produces products during the operation [9,10]. The appropriate manufacturing varies depending upon the characteristics of the pharmaceutical process [8]. For example, batch manufacturing is effective for long processing times and multi-stage processes such as reaction or refining. Continuous manufacturing is suitable for processes with short processing times and high exothermic reactions, such as crystallization or tablet processing [9]. Therefore, many pharmaceutical manufacturing plants require to equip the hybrid batch–continuous manufacturing process that reflects the characteristics of pharmaceutical processes to enhance both productivity and quality.

In this paper, we study an integrated scheduling problem with heterogeneous manufacturing plants and one distribution center (DC). In the two-stage pharmaceutical supply chain, hybrid batch-continuous flowshop manufacturing processes and a direct-shopping delivery are used in response to the flexible response to uncertain demand. The integrated manufacturing and delivery scheduling problem in the two-stage pharmaceutical supply chain is called PS-HF&DS.

In each plant, an unrelated permutation flow shop layout consisting of a hybrid batch-continuous production line is configured to produce various pharmaceuticals. Each pharmaceutical order is split and produced in a distributed manner across the plants. The pharmaceutical products are directly shipped to DC after production. To minimize the makespan, we should (i) split each pharmaceutical order into sub-orders, (ii) assign the sub-orders to plants, and (iii) determine the permutation sequence of production and direct shipment. We formulate PS-HF&DS as a mathematical model. We propose genetic algorithm (GA) and particle swarm optimization (PSO), and two solution structures, each with a unique decoding process. As a result, we develop four variants, comprising two distinct solution structures for each metaheuristic to solve this model. We conduct numerical experiments to verify and evaluate the proposed metaheuristics' performance.

The remainder of our study is organized as follows: In Section 2, we provide the literature survey. In Section 3, we describe the addressed problem and the mathematical formulation. In Section 4, we describe the proposed metaheuristics. In Sections 5 and 6, we show the results of the numerical experiment and sensitivity analysis, respectively. In Section 7, we summarize the conclusion.

2. Literature Survey

The addressed problem is related to the pharmaceutical scheduling and distributed permutation flowshop scheduling (DPFS) problems. We review relevant studies focusing on production systems, manufacturing strategy, scheduling constraints, methodologies, and objective functions. Table 1 summarizes and classifies the reviewed literature.

A real-life pharmaceutical batch scheduling problem formed as a flowshop with multiple processors (FSMP) was addressed in [13]. FSMP considered cleaning, changeover, etc., to minimize total completion time. A hybrid genetic algorithm was proposed and demonstrated robust performance through comparative experiments. The pharmaceutical scheduling problems in a multi-stage batch plant were presented in [14,15]. A MIP-based decomposed heuristic was proposed to solve the scheduling problem and verified by using the benchmark instances of a pharmaceutical facility [14]. To minimize the makespan, they proposed a constraint programming model in [15]. The planning and scheduling of multi-product in the multi-stage continuous plant problem was addressed in [16]. To deal with the addressed problem, three novel MILP models were proposed. A campaign planning and scheduling problem in biopharmaceutical manufacturing was tackled in [11]. The tackled problem considered batch and continuous processes, sequence-dependent changeovers, etc. A mathematical model was developed, and managerial insights were presented by comparison with the previously developed models.

 doi:10.20944/preprints202603.1065.v1

A distributed permutation flowshop scheduling (DPFS) problem, first discussed in [17], has attracted significant attention in the scheduling study. In the last decade, the DPFS problem was extended with variants such as blocking and buffer constraints. The DPFS problem, considering a blocking constraint, was studied in [18,19]. A variant of the iterated greedy algorithm, PBIG, was proposed to minimize total flowtime [18]. To minimize makespan, an evolution strategy (ES) and two mathematical models were proposed in [19]. DPFS with no-wait constraint was addressed in [20]. To minimize the makespan, the general variable neighborhood search (GVNS) algorithm was proposed, and the performance of the proposed GVNS was verified through computational experiments. A distributed permutation heterogeneous flowshop scheduling with order acceptance constraint was studied in [21]. To maximize profit, they proposed GA and PSO. In addition to these studies, DPFS is extended with variants such as setup times constraint [22–24], order constraint [25], integration with transportation [26,27], heterogeneous distributed plants [24,28], distributed hybrid flowshop framework [29,30], and distributed assembly flowshop framework [31–33]. The variants of DPFS have been studied with various methodologies: mathematical model [19,31], iterated greedy algorithm [18,28], variable neighborhood search [20], evolutionary algorithm [19], fruit fly optimization[23], etc. Additionally, variants of the DPFS problem are reviewed and classified in [5,34].

We study the scheduling model simultaneously considering distributed manufacturing, hybrid batch-continuous manufacturing, order splitting, direct shipping policy, and sequence-dependent changeovers to minimize the makespan in the two-stage pharmaceutical supply chain. Despite the practical advantages and necessity of the proposed integrated scheduling in the pharmaceutical industry, to the best of our knowledge, there is no study for the scheduling model related in the pharmaceutical supply chain due to a problem-complexity. Therefore, the contributions of this study are as follows:

(1) We firstly address PS-HF&DS and formulate it as a MILP model to minimize the makespan.
(2) To efficiently solve the large-sized model, we independently apply GA and PSO to two distinct solution representations. As a result, we propose four population-based metaheuristics variants.
(3) We present managerial insights derived from a sensitivity analysis evaluating the performance of the metaheuristics depending upon the maximum algorithm computational time.

Table 1. Summary of literature survey.

STUDY	PS		DM	CO	OS	TR	Constraint	Methodology	OBJ.	
	BP	CP							PROFIT BASED	TIME BASED
[11]	√	√		√			Campaign, Shelf-life, etc.	Mathematical model	√	
[13]	√			√				HGA		√
[14]	√			√				MIP-based decomposition	√	√
[15]	√			√			Campaign, Shelf-life, etc.	CP		√
[16]		√		√				MILP-based heuristic	√	
[17]			√					MILP, VNS		√
[18]			√				Blocking	PBIG		√
[19]			√				Blocking	MIP, CP, ES		√
[20]			√				No-wait	GVNS		√
[21]			√	√			Order acceptance	GA, PSO	√	
[22]			√	√				DABC, Heuristic		√
[23]			√	√				DFFO		√
[24]			√	√	√			MILP, NEABC		√
[25]			√				Order constraint	MILP, ORVND, ORABC, ORIG		√
[26]			√			√		MILP, EBWO		√
[27]			√			√	Multi-objective	MILP, MOBSO	√	√
[28]			√					IG_tb		√
[29]			√				Hybrid flowshop	Constructive heuristics		√
[30]			√				Hybrid flowshop	MILPs		√

 doi:10.20944/preprints202603.1065.v1

[31]			√				Assembly	MILP, Heuristic, VND		√
[32]			√			√	Assembly	Heuristic, VND, IG	√	
[33]			√	√			Assembly	Heuristic, IG		√
This	**√**	**√**	**√**	**√**	**√**	**√**		**MILP, GA, PSO**		**√**

* PS: Pharmaceutical Scheduling; BP: Batch Process; CP: Continuous Process; DM: Distributed Manufacturing; CO: Change-over (Setup); OS: Order Splitting; TR: Transportation; OBJ: Objective Value;.

3. Mathematical Model

We formulate the addressed problem as a mixed-integer-linear-programming (MILP) model. We present the description and an illustrative example of the addressed problem in Section 3.1. In Section 3.2, we present the MILP formulation. The sets, parameters, and variable are defined as follows:

Set & Parameter

DF	Distributed plant set
TB	Batch task set
TC	Continuous task set
T	Total task set
I	Ordered pharmaceutical product set
L	Sub-order set related to pharmaceutical order, $L = I$
ID	Pharmaceutical products and dummy product set, $\{0\} \cup I$
LD	Sub-orders related to pharmaceutical order and dummy set, $LD = ID$
W_i	Order amount of pharmaceutical order $i \in I$
$F1_i$	Type of pharmaceutical order $i \in I$
$F2_l$	Type of production sub-orders $l \in L$
PR_{ti}^f	Production rate of task $t \in T$ for pharmaceutical order $i \in I$ in distributed plant $f \in DF$
Y_{ti}^f	Production yield of task $t \in T$ for pharmaceutical order $i \in I$ in distributed plant $f \in DF$
CO_{ij}^f	Change-over time between pharmaceutical orders $i, j \in ID$ in distributed plant $f \in DF$
DT_f	Delivery time between distributed plant $f \in DF$ and DC
M	Large number

Decision Variable

z_{fl}	Equal to 1, if sub-order $l \in L$ in distributed plant $f \in DF$ is assigned; Otherwise, 0;
pq_{tl}^f	Production amount of sub-order $l \in L$ at task $t \in T$ in distributed plant $f \in DF$
q_{fl}	Production amount of sub-order $l \in L$ in distributed plant $f \in DF$
x_{ij}^f	Equal to 1, if sub-order $j \in LD$ is immediately preceded sub-order $i \in LD$ in distributed plant $d \in DF$; Otherwise, 0;
um_l^f	Manufacturing sequence of sub-order $l \in LD$ in distributed plant $f \in DF$
st_{tl}^f	Start time of sub-order $l \in LD$ at task $t \in T$ in distributed plant $f \in DF$
pt_{tl}^f	Productions time of sub-order $l \in LD$ at task $t \in T$ in distributed plant $f \in DF$
ct_{tl}^f	Completion time of sub-order $l \in LD$ at task $t \in T$ in distributed plant $f \in DF$
cm_l^f	Manufacturing completion time of sub-order $l \in L$ in distributed plant $f \in DF$
dc_l^f	Delivery completion time of sub-order $l \in L$ in distributed plant $f \in DF$
C_i	Completion of pharmaceutical order $i \in I$
C_{max}	Makespan

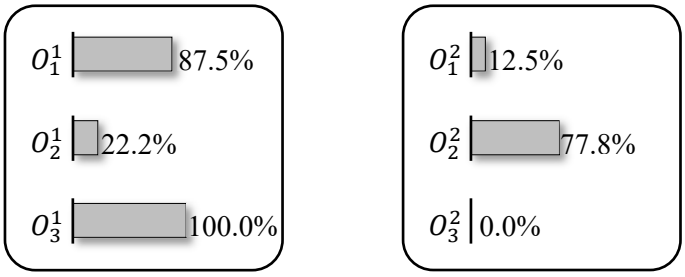
3.1. Problem Description

We assume that DC has the same number of vehicles as the number of plants to pick up the produced pharmaceutical products from the plants. The detailed descriptions of the addressed PS-HF&DS are as follows:

 doi:10.20944/preprints202603.1065.v1

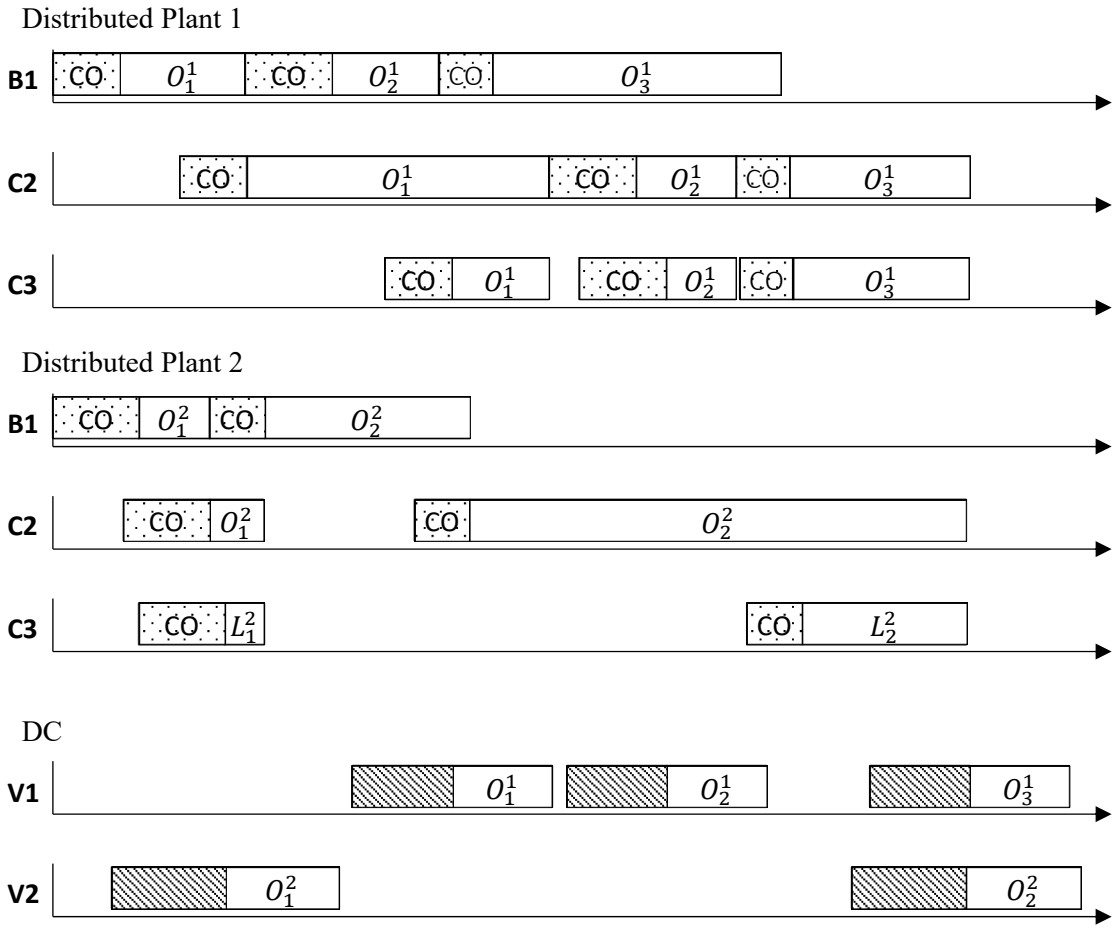
(1) Each pharmaceutical order is split into sub-orders, and the production amount of each sub-order is determined in the order splitting phase. The sum of the production amount of the sub-orders must be equal to the order amount of the related order.
(2) The sub-orders are assigned to plants in the sub-order assigning phase. The processing time of each sub-order differs at each task depending upon the distributed plant, and it is calculated based on [16].
(3) In the permutation sequencing of production and direct shipment phase, we determine not only the permutation sequence but also the start time and the completion time of each sub-order at each task in the distributed plant. Between two consecutive continuous-task stages, for each sub-order, the next task starts after the previous task starts, and the next task completes after the previous task completes [16].
(4) The DC picks up the pharmaceutical products according to a direct shipping policy. Since vehicle fleet utilization and routing are not within the scope of this study, the products are directly delivered without allowing order-splitting and combining. The completion of the pharmaceutical order is defined as the latest arrival time at the DC among all related sub-orders.

We illustrate the PS-HF&DS in Figure 1 with three pharmaceutical orders and two distributed plants layouts as one batch task and two consecutive continuous task stages. Figure 1 (a) represents the result of the order splitting and assigning, where bar charts represent the production amount of each sub-order in every plant as a ratio. For bar charts of Figure 1 (a), the Y-axis represents the sub-orders in each plant. Index O_i^f means the sub-order for the pharmaceutical order i produced in the plant f. The X-axis represents the amount produced by each plant as a percentage of the total order amount. We show the Gantt charts for plants and DC in Figure 1 (b). The permutation manufacturing sequences of the sub-orders correspond to $[O_1^1 \rightarrow O_2^1 \rightarrow O_3^1]$ and $[O_1^2 \rightarrow O_2^2]$ in the D-Plants 1 and 2, respectively. For each distributed plant, since stages C2 and C3 are two consecutive continuous task stages, each sub-order begins task C3 before task C2 is finished. However, each sub-order conducts task C2 only after task B1 is finished. In the shipping stage, each sub-order is directly delivered to the DC after production. The completion time of each order is defined as the latest arrival time of the sub-order related to the order.



(a) Result of order splitting

 doi:10.20944/preprints202603.1065.v1



(b) Gantt charts of distributed plants and DC

Figure 1. Illustrative example of HPS-DPSC.

3.2. Mixed-Integer-Linear-Programming Model

The objective function and constraints of MILP model can be formulated as follows:

$$Minimize\ C_{max} \tag{1}$$

Subject to

$$q_{fl} \le M \times z_{fl} \qquad \forall f \in DF, l \in L \tag{2}$$

$$W_i \le \sum_{f \in DF} q_{fl} \qquad \forall i \in I, l \in L: F1_i = F2_l \tag{3}$$

$$x_{ii}^f = 0 \qquad \forall f \in DF,\ i \in LD \tag{4}$$

$$x_{0i}^f + \sum_{j \in L: i \ne j} x_{ji}^f = z_{fl} \qquad \forall f \in DF,\ i \in L \tag{5}$$

$$x_{i0}^f + \sum_{j \in L: i \ne j} x_{ij}^f = z_{fl} \qquad \forall f \in DF,\ i \in L \tag{6}$$

$$\sum_{i \in L} z_{fi} \le M \times \sum_{j \in L} x_{0j}^f \qquad \forall f \in DF \tag{7}$$

$$\sum_{i \in L} z_{fi} \le M \times \sum_{j \in L} x_{j0}^f \qquad \forall f \in DF \tag{8}$$

$$um_0^f = 0 \qquad \forall f \in DF \tag{9}$$

$$um_i^f \le M \times z_i^f \qquad \forall f \in DF,\ i \in L \tag{10}$$

$$um_i^f \le \sum_{j \in L} z_j^f \qquad \forall f \in DF,\ i \in L \tag{11}$$

$$um_i^f + x_{ij}^f \le um_j^f + M \times \left(1 - x_{ij}^f\right) \qquad \forall f \in DF, i \in LD, j \in L: i \ne j \tag{12}$$

$$pq_{tl}^f = pt_{tl}^f \times PR_{ti}^f \qquad \forall f \in DF, i \in I, l \in L: F1_i = F2_l,\ t \in T \tag{13}$$

$$pq_{tl}^f \times Y_{tl}^f = pq_{t+1,l}^f \qquad \forall f \in DF, i \in I, l \in L: F1_i = F2_l,\ t \in T: T \ne |T| \tag{14}$$

$$pq_{tl}^f \times Y_{ti}^f = q_{fl} \qquad \forall f \in DF, i \in I, l \in L: F1_i = F2_l,\ t \in T: T = |T| \tag{15}$$

 doi:10.20944/preprints202603.1065.v1

$$st_{tl}^{f} \le M \times z_{fl} \qquad \forall f \in DF,\ l \in L,\ t \in T \qquad (16)$$
$$ct_{tl}^{f} \le M \times z_{fl} \qquad \forall f \in DF,\ l \in L,\ t \in T \qquad (17)$$
$$st_{tl}^{f} + pt_{tl}^{f} = ct_{tl}^{f} \qquad \forall f \in DF,\ l \in L,\ t \in T \qquad (18)$$
$$ct_{ti}^{f} + CO_{ij}^{f} \le st_{tj}^{f} + M \times \left(1 - x_{ij}^{f}\right) \qquad \forall f \in DF, i, j \in L : i \ne j, t \in T \qquad (19)$$
$$CO_{ti}^{f} \le st_{ti}^{f} + M \times \left(1 - x_{0i}^{f}\right) \qquad \forall f \in DF, i \in L : i \ne j, t \in T \qquad (20)$$
$$ct_{tl}^{f} \le st_{t+1,\,l}^{f} + M \times \left(1 - z_{fl}\right) \qquad \forall f \in DF,\ l \in L,\ t \in TB \qquad (21)$$
$$st_{il}^{f} \le st_{jl}^{f} + M \times \left(1 - z_{fl}\right) \qquad \forall f \in DF, l \in L : i, j \in TC : i < j \qquad (22)$$
$$ct_{il}^{f} \le ct_{jl}^{f} + M \times \left(1 - z_{fl}\right) \qquad \forall f \in DF, l \in L : i, j \in TC : i < j \qquad (23)$$
$$ct_{tl}^{f} \le cm_{l}^{f} + M \times \left(1 - z_{fl}\right) \qquad \forall f \in DF, l \in L,\ t \in T : t = |T| \qquad (24)$$
$$cm_{l}^{f} + DT_{f} \le dc_{l}^{f} + M \times \left(1 - z_{fl}\right) \qquad \forall f \in DF, l \in L \qquad (25)$$
$$dc_{i}^{f} + 2 \times DT_{f} \le dc_{j}^{f} + M \times \left(1 - x_{ij}^{f}\right) \qquad \forall f \in DF, i, j \in L : i \ne j \qquad (26)$$
$$cm_{l}^{f} + DT_{f} \le dc_{l}^{f} + M \times \left(1 - x_{0l}^{f}\right) \qquad \forall f \in DF, l \in L \qquad (27)$$
$$dc_{l}^{f} \le C_{i} \qquad \forall f \in DF, i \in I, l \in L : F1_{i} = F2_{l} \qquad (28)$$
$$C_{i} \le C_{max} \qquad \forall i \in I \qquad (29)$$
$$z_{fl}, x_{ij}^{f} \in \{0, 1\} \qquad \forall f \in DF, l \in L, j \in LD, j \in LD \qquad (30)$$
$$pq_{tl}^{f}, q_{fl}, cm_{l}^{f}, dc_{l}^{f} \ge 0 \qquad \forall f \in DF, l \in L, t \in T \qquad (31)$$
$$st_{tl}^{f}, pt_{tl}^{f}, ct_{tl}^{f} \ge 0 \qquad \forall l \in LD, t \in T, f \in DF \qquad (32)$$
$$C_{i} \ge 0 \qquad \forall i \in I \qquad (33)$$

Constraint (1) represents the objective function of the proposed PS-HF&DS to minimize the makespan. Constraints (2) and (3) determine the production amount of each sub-order in every plant. The permutation manufacturing sequence in each distributed plant is determined by constraints (4) to (12). The production sequence of the sub-orders i and j in the distributed plant f are determined by in Constraints (5) and (6). Constraints (7) - (8) ensure that if there is any sub-order i assigned to the distributed plant f, the first and last manufacturing sequence order must exist in the distributed plant f. Constraints (9) – (12) prevent sub-tours in distributed plant f. Constraints (13) – (24) represent the permutation sequencing phase. Constraints (13) – (15) calculate the production time of the sub-order l in each task and its production amount based on [35]. Constraints (16) – (18) represent the relationship among start, production, and completion times of sub-order l. Constraints (19) – (20) calculate start and completion times of sub-orders i and j within a task. Constraint (21) calculates start and completion times between consecutive batch tasks t and $t+1$. Constraints (22) and (23) calculate start and completion times of sub-order l between continuous tasks t and $t+1$. The manufacturing completion time of sub-order l in distributed plant f is calculated by constraint (24). Constraints (25) – (29) represent the shipping stage. Constraints (25) and (26) calculate the delivery completion time of sub-order l in distributed plant f. For each plant f, the first manufactured sub-order l is delivered and calculated the delivery completion time by Constraint (27). The completion time of pharmaceutical sub-order i and makespan are calculated by constraint (28) and (29), respectively. Constraints (30) – (33) are binary and non-negative constraints for decision variables.

4. Metaheuristics

Since DPFS problem has been proven NP-hard problem [25], the large-sized proposed problem is not able to be solved within a limited time by the MILP model. Hence, we propose genetic algorithm (GA) and particle swarm optimization (PSO) to solve and find the optimal solutions of the reasonable computational time. GA and PSO have demonstrated good performance in related studies [36–40]. We introduce two distinct solution structures with unique decoding processes. The first structure determines order splitting, assignment, and permutation sequence through solution structure in the decoding process, and we refer to a metaheuristic using this solution structure as *_OFP. The second structure one also determines order splitting and permutation sequence by solution structure. However, the plant assignment is determined via a completion time-based

 doi:10.20944/preprints202603.1065.v1

assignment heuristic (CAH) in the decoding process. We refer to a metaheuristic using this solution structure as *_OP-CAH. Consequently, we develop four variants, GA_OFP, GA_OP-CAH, PSO_OFP, and PSO_OP-CAH, and these four variants are classified in Figure 2.

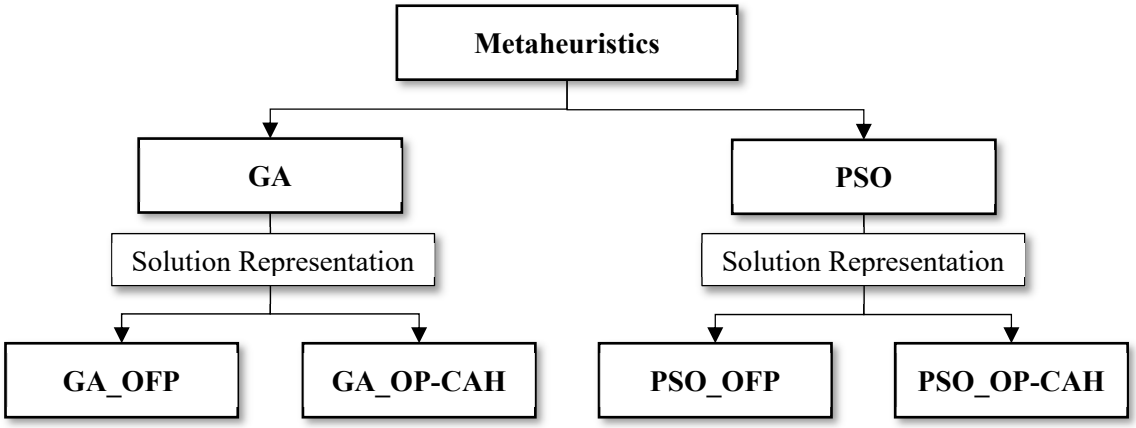


Figure 2. Classification of metaheuristics.

*4.1. Solution Structure and Decoding PROCESS for *_OFP*

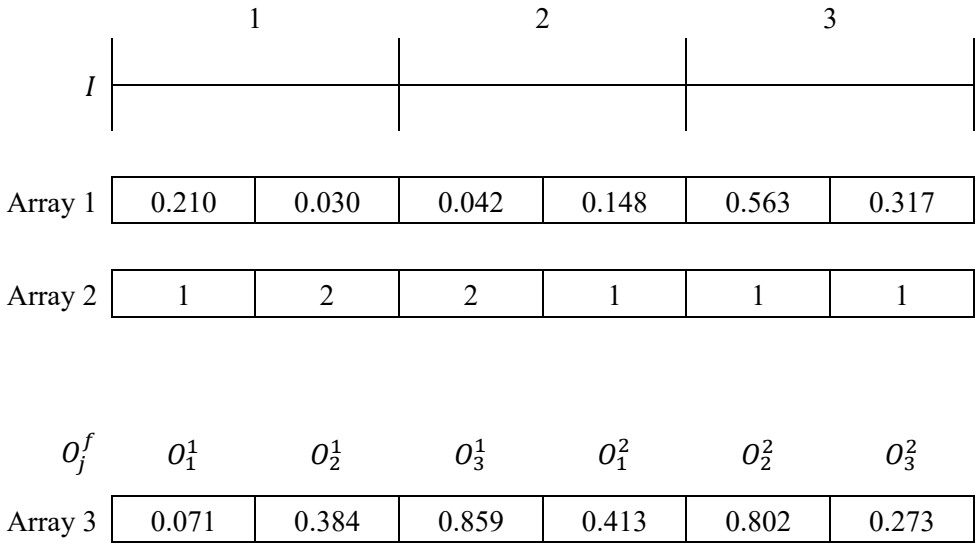
*_OFP uses three one-dimensional arrays as an encoded solution. Arrays 1 and 3 are represented as decimals generated by following a uniform distribution $U(0,1)$ with $|DF| \times |I|$ length. Array 2 is represented as an integer generated by following a discrete uniform distribution $unif\{1, |DF|\}$ with $|DF| \times |I|$ length. An example of an encoded solution is illustrated in Figure 3. Figure 3 (a) shows an encoded solution using the data from Figure 1. Arrays 1, 2, and 3 correspond to [0.210, 0.030, 0.042, 0.0148, 0.563, 0.317], [1, 2, 2, 1, 1, 1], and [0.071, 0.384, 0.859, 0.413, 0.802, 0.273], respectively.

Array 1 splits each pharmaceutical order into $|DF|$ virtual sub-orders. It also calculates the production amount of each virtual sub-order by normalizing the decimal elements of Array 1 related to order i. Each virtual sub-order is assigned to a plant based on the integer elements of Array 2. The actual sub-orders and production amount of the sub-order related to order i are determined for each plant via decoding Arrays 1 and 2. Figure 3 (b) illustrates the process of the order splitting using Array1. In Figure 3 (b), Array 1 is decoded as the production percentage array. Array 1 [0.210, 0.030, 0.042, 0.148, 0.563, 0.317] becomes $\left[\frac{0.210}{0.210+0.030}, \frac{0.030}{0.210+0.030}, \frac{0.148}{0.148+0.042}, \frac{0.042}{0.148+0.042}, \frac{0.563}{0.563+0.317}, \frac{0.317}{0.563+0.317}\right]$ as [87.5 (%), 12.5 (%), 77.8 (%), 22.2 (%), 64.0 (%), 36.0 (%)]. Since the elements 0.210 and 0.030 are related to pharmaceutical order 1, these elements are normalized as $\frac{0.210}{0.210+0.030}$ and $\frac{0.030}{0.210+0.030}$, respectively. Figure 3 (c) illustrates assigning each virtual sub-order into plant f using Array 2. For pharmaceutical order 3, since every virtual sub-order is assigned to the distributed plant 1, the production amount of the actual sub-order related to order 3 in the distributed plant 1, q_{13}, is represented as 100.0 (%).

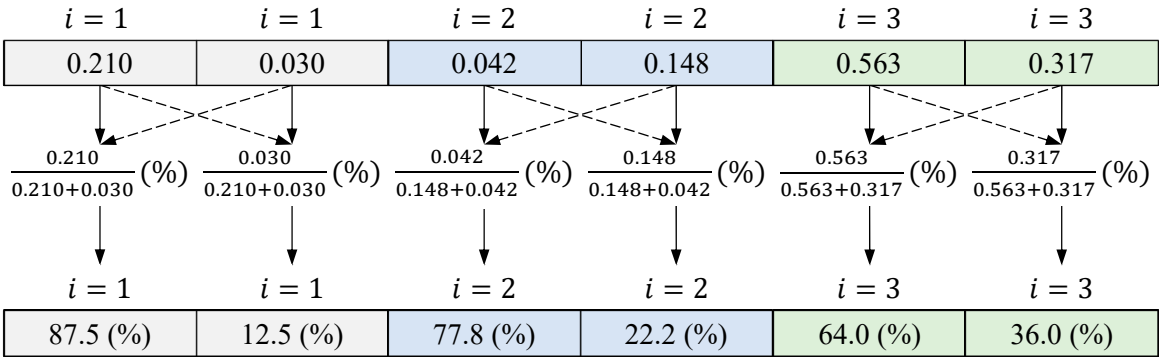
Array 3 indicates the production and direct shipping sequence of the sub-order O_i^f by sorting the elements of Array 3 in increasing order. Each sub-order O_i^f is produced at the earliest possible start time st_{tl}^f in each task t in distributed plant f. Figure 3 (d) illustrates the production and direct shipping sequence of each sub-order within a distributed plant corresponding to the Gantt chart in Figure 1 (a).

 doi:10.20944/preprints202603.1065.v1

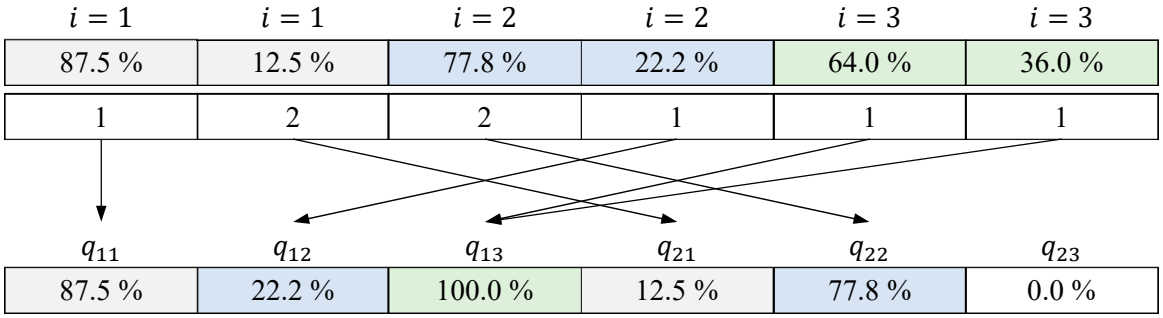
(a) Encoded solution structure



*O_i^f: sub-order related to pharmaceutical order $i \in I$ produced in plant $f \in F$.



(b) Determining process of order splitting with Array 1



*q_{fl}: Production amount of sub-order related to order $l \in I$ produced in plant $f \in F$.

(c) Determining process of order assignment

DF 1	O_1^1 (87.5 %)	O_2^1 (22.2 %)	O_3^1 (100.0 %)
DF 2	O_1^2 (12.5 %)	O_2^2 (77.8 %)	

(d) Production and direct shipping sequence

Figure 3. Solution representation and decoding process of OFP algorithm.

 doi:10.20944/preprints202603.1065.v1

*4.2. Solution Structure and Decoding Process for *_OP-CAH*

*_OP-CAH uses two one-dimensional arrays as an encoded solution. Arrays 1 and 2 are represented as decimals generated following a uniform distribution $U(0,1)$ with $|DF| \times |I|$ length. The production amount of each virtual sub-order is determined by decoding Array 1 as illustrated in Figure 3 (b). Array 2 indicates the assigning sequence of the virtual sub-orders by sorting its elements in increasing order. We determine the sub-order assignment by using a completion time-based assignment heuristic (CAH). In CAH, the current dispatching sub-order is virtually assigned to all plants and then assigned into the plant with the minimum delivery completion time. The pseudocode of CAH is as follows:

Pseudo code 1: Completion time-based assignment heuristic

Input: Order List of the sub-orders

Output: C_{max}

Begin
  Let $O_{[x]}$ as xth sorted virtual sub-order in the Order List
  Let i^* as pharmaceutical order related to $O_{[x]}$
  Let $q_{[x]}$ as the production amount of $O_{[x]}$
  Let dispatching index $x \leftarrow 1$
  For $(1 \leq x \leq |DF| \times |I|)$
    For $(f \in DF)$
      Let $\widehat{q_{fi^*}}$ as virtual production amount of sub-order related to order i^* in plant f
      If order i^* is not assigned into distributed plant f
        $\widehat{q_{fi^*}} \leftarrow q_{[x]}$
      Else
        $\widehat{q_{fi^*}} \leftarrow q_{fi^*} + q_{[x]}$
      End If
      Let Vdc_f virtual delivery completion time of $O_{[x]}$ when the sub-order is assigned into distributed plant f
      Calculate Vdc_f based on $\widehat{q_{fi^*}}$
    End For
    Assign $O_{[x]}$ into $f^* \leftarrow \underset{f \in DF}{\operatorname{argmin}} Vdc_f$ and update $dc_{i^*}^{f^*} \leftarrow \min_{f \in DF} Vdc_f$
    From $dc_{i^*}^{f^*}$, backwardly update st_{tl}^{f} and ct_{tl}^{f}, $\forall\, t \in T$
    Update $q_{f^*i^*} \leftarrow q_{f^*i^*} + q_{[x]}$
  End For
  Calculate C_{max} following Constraints (28) – (29)
End

4.3. Genetic Algorithm

GA utilizes a set of encoded solutions called chromosomes as a population. It consists of several processes: initialization, decoding, evaluation, selection, and crossover and mutation. A population is initially created using distribution $U(0,1)$ and $unif\{1,|DF|\}$ in the initialization process. In the decoding process, to obtain C_{max}, each chromosome is decoded by following Sections 4.1 and 4.2. The fitness of each chromosome i, $F(i)$ is calculated by Eq. (34) in the evaluation process.

 doi:10.20944/preprints202603.1065.v1

$$F(i) = {}^{1}/_{C_{max}^{i}} \quad (34)$$

, where C_{max}^{i} denotes C_{max} obtained by chromosome i. In the selection process, we adopt a selection mechanism, the roulette wheel method.. For the crossover operation, a one-cut crossover is conducted. For the mutation operation, Arrays 1 and 2 are tweaked by uniform and swap mutations, respectively. Figure 4 indicates the flow charts of the proposed GA.

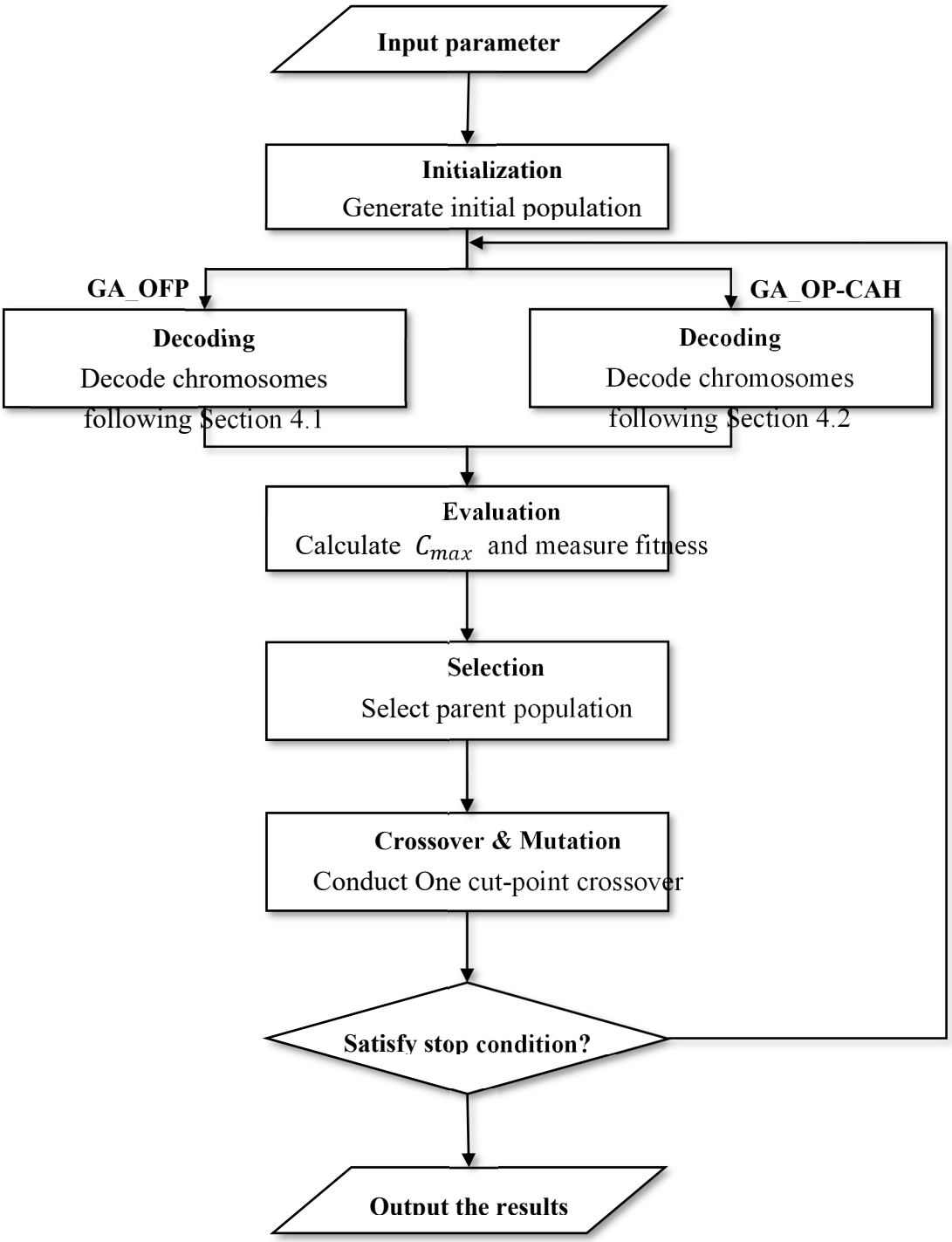


Figure 4. Flow chart of GA_ OFP and GA_OP-CAH.

 doi:10.20944/preprints202603.1065.v1

.4.4. Particle Swarm Optimization

PSO utilizes a set of encoded solutions called particles as a swarm. It consists of several processes: initialization, decoding, evaluation, best position update, and velocity and position update. The initialization, decoding, and evaluation processes of PSO are identical with GA. In the best position update process, the personal and global best position vectors are updated by following Eqs.(35) - (36)

$$P_p^t \leftarrow \begin{cases} X_p^t, & F(X_p^t) > F(P_p^t) \\ P_p^t, & otherwise \end{cases} \tag{35}$$

$$G^t \leftarrow \begin{cases} X_p^t, & F(X_p^t) > F(G^t) \\ G^t, & otherwise \end{cases} \tag{36}$$

, where X_p, P_p, and G represent position and personal best position vectors of particle p and global best position vector in iteration t, respectively. Each array of the position and velocity vectors are independently updated by following Eqs. (37) – (39) in the velocity and position update processes, respectively. The vectors represented as $U(0,1)$ are updated by following Eqs. (37) - (38)

$$V_p^{t+1} \leftarrow w \times V_p^t + c_1 \times U(0,1) \times (P_p^t - X_p^t) + c_2 \times U(0,1) \times (G^t - X_p^t) \tag{37}$$

$$X_p^{t+1} \leftarrow X_p^t + V_p^{t+1} \tag{38}$$

, where w, c_1, and c_2 represent weight, cognitive coefficient and social coefficient, respectively. V_p^{t+1} represents velocity vector in iteration t. The position vectors represented as $unif\{1, n_{DF}\}$ are updated by following Eq. (39) referring pan et al. [41]

$$X_p^{t+1} \leftarrow c_2 \otimes F_3(c_1 \otimes F_2(w \otimes F_1(X_p^t), P_p^t), G^t) \tag{39}$$

, where F_1, F_2, and F_3 are mutation and crossover operators [41]. The pseudocode of Eq. (39) is as follows:

Pseudo code 2: The position vector of the Array 2 update process
Input: Position vector of the Array 2 X_p^t, weight w, cognitive coefficient c_1, social coefficient c_2
Output: Updated position vector of the Array 2 X_p^{t+1}

Begin
  $r_1, r_2, r_3 \leftarrow U(0,1)$
  If $r_1 < w$
    $\lambda_p^{t+1} \leftarrow mutation(X_p^t)$
  Else
    $\lambda_p^{t+1} \leftarrow X_p^t$
  End If
  If $r_2 < c_1$
    $\delta_p^{t+1} \leftarrow crossover(\lambda_p^{t+1}, P_p^t)$
  Else
    $\delta_p^{t+1} \leftarrow P_p^t$
  End If
  If $r_3 < c_2$
    $X_p^{t+1} \leftarrow crossover(\delta_p^{t+1}, G^t)$
  Else
    $X_p^{t+1} \leftarrow \delta_p^{t+1}$
  End If
End

, where $mutation$ and $crossover$ mean swap mutation and one-cut crossover, respectively.

Figure 5 indicates the flow charts of the proposed PSO.

 doi:10.20944/preprints202603.1065.v1

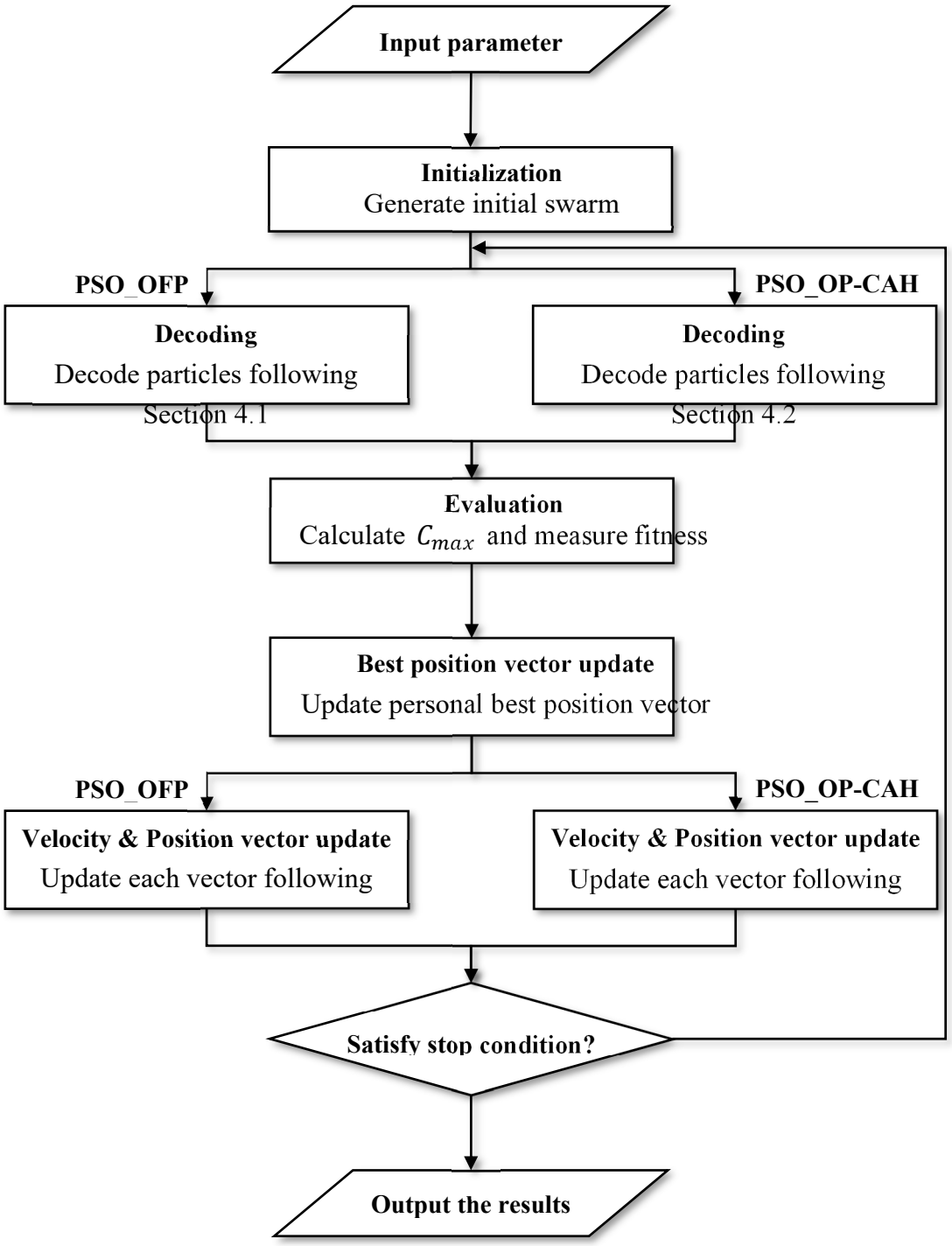


Figure 5. Flow chart of PSO_ OFP and PSO_OP-CAH.

5. Experiments

We conduct numerical experiments to validate the proposed metaheuristics' performance. The complexity of PS-HF&DS is affected by the number of virtual sub-orders, which is the interaction between the number of ordered pharmaceutical products and distributed plants, $|DF| \times |I|$. Therefore, depending upon the size of $|DF|$ and $|I|$, we generate two types of instances, which are small-sized and large-sized. $|DF|$ is $\{2, 3\}$, and $|I|$ is $\{6, 8, 10\}$ in the small-sized instances. $|DF|$ is $\{4, 5, 6\}$, and $|I|$ is $\{40, 50, 60\}$ in the large-sized instances.

To reflect the features of the pharmaceutical supply chain, we generate the experimental parameters by referring to other related studies. For the manufacturing process layouts, we adopt

 doi:10.20944/preprints202603.1065.v1

three flowshop layouts as $\{(C,C),(B,C,C),(B,C,C,B)\}$ presented in [11], in which B and C represent batch and continuous tasks, respectively. Since no study simultaneously considers sequence-dependent change-over time, batch and continuous tasks, and distributed manufacturing, we randomly generate change-over time, production rate, and production yield by referring to pharmaceutical scheduling studies [11,14,16]. In addition, we determine weight of pharmaceutical order, W_i, by discrete uniform distribution $unif\{|DF|, 3\times|DF|\}$ and the delivery time between distributed plant and DC as *Manhattan distance*.

For every instance, we replicate the proposed metaheuristics 30 times with the maximum computational time as $|DF|\times|I|\times 0.5$. The performance measures of small and large size instances are absolute and relative percentage deviation ($ARPD$) as follows:

$$ARPD_i\ (\%) = \frac{best - alg_i}{best}\times 100, \tag{40}$$

, where alg_i and $best$ are solution obtained by metaheuristic i and the best solution, respectively. Within the limited time, if the MILP model finds the optimal solutions of the small-sized instances, $best$ means optimal solution. MILP model and metaheuristics are implemented by IBM ILOG CPLEX 22.1.1 and Python IDE, respectively. We use experimental environment as an Intel Core i7-13700KF 3.40GHz CPU.

5.1. Parameter Calibration

We calibrate metaheuristic parameters by using the two-phase calibration method [42,43]. For GA_OFP and GA_OP-CAH, the popsize (P) and crossover and mutation rates (CR, MR) are calibrated. For PSO_OFP and PSO_OP-CAH, the swarm size (S), weight (w), and cognitive and social coefficients (c_1, c_2) are calibrated. Every metaheuristic metaheuristic use the stopping criterion as a maximum CPU time, $|DF|\times|I|\times 0.5$ seconds. Especially, we set the parameters P and S as $|DF|\times|I|\times x$, and we determine x in the two-phase calibration. To calibrate the parameters, three experimental instances are randomly generated and run five replications.

Table 2 shows the result of the parameter calibration with the best parameters of each metaheuristic in bold. For each parameter, if the best value is the largest or smallest candidate in the first calibration phase, we conduct the second phase to finely tune the parameter. For the GA_OFP, P, CR, and MR are 1.00, 0.70, and 0.15. For the GA_OP-CAH, P, CR, and MR are 1.00, 0.40, and 0.15. For PSO_OFP, S, w, c_1, and c_2 are 0.75, 0.45, 0.20, and 0.30. For PSO_OP-CAH, S, w, c_1, and c_2 are 1.00, 0.40, 0.15, and 0.50. We set the termination condition of every metaheuristic as the maximum computational time, $|DF|\times|I|\times 0.5$ seconds.

Table 2. Calibration of algorithm parameters.

Metaheuristic	Parameter	First Calibration	Second Calibration
GA_OFP	P	0.50, **1.00**, 1.50	-
	CR	0.40, 0.50, **0.60**	0.60, **0.70**
	MU	0.10, **0.15**, 0.20	-
GA_OP-CAH	P	0.50, **1.00**, 1.50	-
	CR	0.20, 0.30, **0.40**	**0.40**, 0.50
	MU	0.05, 0.10, **0.15**	**0.15**, 0.20
PSO_OFP	S	**0.75**, 1.00, 1.25	0.50, **0.75**
	W	0.15, 0.25, **0.35**	0.35, **0.45**
	c_1	**0.20**, 0.30, 0.40	0.10, **0.20**
	c_2	0.15, 0.20, **0.25**	0.25, **0.30**
PSO_OP-CAH	S	0.50, **1.00,** 1.50	-
	W	0.10, 0.20, **0.30**	0.30, **0.40**
	c_1	**0.20**, 0.25, 0.30	**0.15**, 0.20
	c_2	0.30, 0.35, **0.40**	0.40, **0.50**

 doi:10.20944/preprints202603.1065.v1

5.2. Experiment Results

In Table 3, the results of the small-sized experiments are shown with the median of $ARPD$ and computational time (CPU). We validate the proposed four metaheuristics with the MILP model described in Section 3.2. We limit the maximum computational time of the MILP model to 1,800 seconds. If the MILP model does not solve the instance within 1,800 seconds, we mark the OBJ and CPU time as the best feasible solution obtained by CPLEX IBM ILOG and 1,800++, respectively. Otherwise, we mark the OBJ as the optimal solution with a star symbol. The averages of median $ARPD$ of GA_OFP, PSO_OFP, GA_OP-CAH, and PSO_OP-CAH are 3.43, 4.72, 1.94, and 3.15, respectively. Every proposed metaheuristic shows the average of median $ARPD$ within 5.0%.

Table 3. Median $\boldsymbol{ARPD}$ results of small-sized experiments.

Ins.	Flowshop Layout	$\|DF\|$	$\|I\|$	$best$	MILP		*_OFP		*_OP-CAH	
					OBJ	CPU time	GA_OFP	PSO_OFP	GA_OP-CAH	PSO_OP-CAH
1	(C,C)	2	6	775.75	**775.75***	21.40	3.09	3.09	1.49	0.54
2			8	905.78	932.33	1800++	3.26	3.26	1.83	2.90
3			10	1180.00	1180.00	1800++	3.15	3.15	1.82	3.14
4		3	6	551.31	**551.31***	1772.86	4.07	5.38	0.00	0.00
5			8	897.20	917.43	1800++	2.48	3.46	2.25	2.96
6			10	1116.71	1177.71	1800++	2.96	7.59	3.38	7.27
7	(B,C,C)	2	6	1166.22	**1166.22***	27.28	1.57	1.84	0.97	1.02
8			8	1128.22	1128.22	1800++	2.92	4.96	1.52	1.89
9			10	1688.84	1714.11	1800++	3.33	3.89	1.12	2.00
10		3	6	796.67	802.53	1800++	2.14	2.80	0.54	2.32
11			8	1024.54	1025.19	1800++	2.74	5.13	2.10	3.49
12			10	969.67	1076.16	1800++	4.98	9.03	3.38	5.91
13	(B,C,C,B)	2	6	785.69	**785.69***	19.27	0.20	0.60	0.00	0.13
14			8	988.99	1030.32	1800++	7.37	7.48	2.79	4.65
15			10	1375.41	1455.89	1800++	6.23	6.23	2.25	4.43
16		3	6	993.00	1002.67	1800++	3.73	4.45	3.41	4.41
17			8	1225.88	1259.31	1800++	3.29	6.05	3.22	4.71
18			10	1473.00	1555.06	1800++	4.24	6.62	2.82	4.89
			Avg.	1057.94	1085.33	1506.71	3.43	4.72	1.94	3.15

* Terminate condition: Maximum CPU time, $|DF| \times |I| \times 0.5$ seconds.

In Table 4, the results of the large-sized experiments are shown with the median of $ARPD$. We compare the proposed metaheuristics by using the $ARPD$. The averages of median $ARPD$ of GA_OFP, PSO_OFP, GA_OP-CAH, and PSO_OP-CAH are 25.87, 16.93, 3.16, and 2.93, respectively. *_OP-CAHs show a lower average of median $ARPD$ than *_OFPs. Within the OP-CAH and *_OFPs, PSO also shows a lower average of median $ARPD$ than GA.

Table 4. Results of large-sized experiments.

Ins.	Flowshop Layout	$\|DF\|$	$\|I\|$	$best$	*_OFP				*_OP-CAH			
					GA_OFP		PSO_OFP		GA_OP-CAH		PSO_OP-CAH	
					$ARPD$	CPU	$ARPD$	CPU	$ARPD$	CPU	$ARPD$	CPU
1	(C,C)	4	40	3629.20	10.95	80.11	5.73	80.06	4.32	82.95	3.82	83.61
2			50	3596.56	18.49	100.11	9.40	100.09	3.58	109.53	4.06	105.58
3			60	5425.04	25.76	120.24	15.95	120.11	2.89	139.23	3.08	124.24
4		5	40	4097.71	20.78	100.10	13.30	100.11	4.52	109.71	4.13	103.56
5			50	4342.04	27.48	125.21	18.40	125.15	4.42	134.44	4.20	143.40
6			60	5428.44	42.91	150.28	26.09	150.27	3.67	151.15	2.88	157.43
7		6	40	3425.20	30.39	120.17	22.66	120.12	4.48	134.18	3.65	122.23
8			50	4245.44	46.28	150.32	35.09	150.14	2.99	176.66	2.96	185.54
9			60	5036.22	58.78	180.43	41.74	180.27	4.18	233.87	3.94	181.25
10	(B,C,C)	4	40	3717.69	10.76	80.12	7.75	80.07	3.52	82.25	3.19	86.55
11			50	4257.79	21.63	100.15	15.24	100.08	3.65	110.45	3.50	113.63
12			60	6434.78	11.62	120.22	5.70	120.17	2.04	144.15	2.02	121.17
13		5	40	5005.33	12.62	100.15	7.23	100.12	2.20	107.92	2.00	114.05
14			50	6617.00	18.65	125.28	10.74	125.20	1.92	155.42	1.98	127.22

 doi:10.20944/preprints202603.1065.v1

15			60	5183.12	41.85	150.37	23.14	150.26	3.37	181.43	3.38	190.43
16		6	40	4231.53	22.06	120.26	16.05	120.13	2.20	142.04	1.94	148.32
17			50	4423.38	49.01	150.26	35.71	150.28	3.17	178.76	3.28	183.06
18			60	6557.32	46.84	180.53	29.88	180.29	4.02	223.11	3.62	226.76
19	(B, C, C, B)	4	40	4056.59	8.97	80.13	5.31	80.08	4.09	87.93	4.22	83.91
20			50	4982.75	16.38	100.12	9.32	100.11	2.87	112.21	2.22	116.58
21			60	7173.84	16.49	120.27	9.14	120.19	3.25	138.74	3.05	144.54
22		5	40	5154.94	10.76	100.14	6.13	100.12	1.62	108.63	1.17	113.19
23			50	5093.44	22.84	125.26	14.93	125.21	2.59	139.79	1.84	145.58
24			60	6533.71	22.71	150.38	13.33	150.24	2.05	186.31	1.86	190.40
25		6	40	4219.74	21.36	120.20	14.45	120.15	2.53	133.65	2.17	137.99
26			50	5271.06	22.55	150.40	17.31	150.34	2.68	188.01	2.54	193.45
27			60	6457.59	39.58	180.74	27.31	180.37	2.54	198.81	2.29	197.48
			Avg.	**4985.09**	**25.87**	**125.26**	**16.93**	**125.17**	**3.16**	**144.12**	**2.93**	**142.26**

* Terminate condition: Maximum CPU time, $|DF| \times |I| \times 0.5$ seconds.

We conduct the ANOVA test with a 0.95 confidence level and present the results as the interval plot in Figure 6. There are no overlaps with the intervals of GA_OFP and PSO_OFP. For the intervals of GA_OP-CAH and PSO_OP-CAH, these do not overlap with the intervals of GA_OFP and PSO_OFP. However, the intervals of GA_OP-CAH and PSO_OP-CAH overlap. It means that *_OFP and *_OP-CAH are statistically significant, and GA and PSO show no statistical significance within the *_OP-CAH. To analyze the robustness of the proposed metaheuristics depending upon the changes in the experimental parameters, we present the interval plots with a 0.95 confidence level for the flowshop layout, $|DF|$, and $|I|$ in Figure 7. In Graphs (a) to (c), GA_OFP and PSO_OFP exhibit significant fluctuation in the mean value and confidence interval width depending upon the experimental parameter changes. Especially, the performance of the *_OFP decreases with a decrease in the number of batch tasks and increases in and $|I|$. In contrast, GA_OP-CAH and PSO_OP-CAH show robust mean values and tight confidence intervals depending upon the experimental parameter changes. Since *_OP-CAH determines the sub-order assignment with the CAH, unlike *_OFP, *_OP-CAH shows robust performance despite the experimental parameter changes.

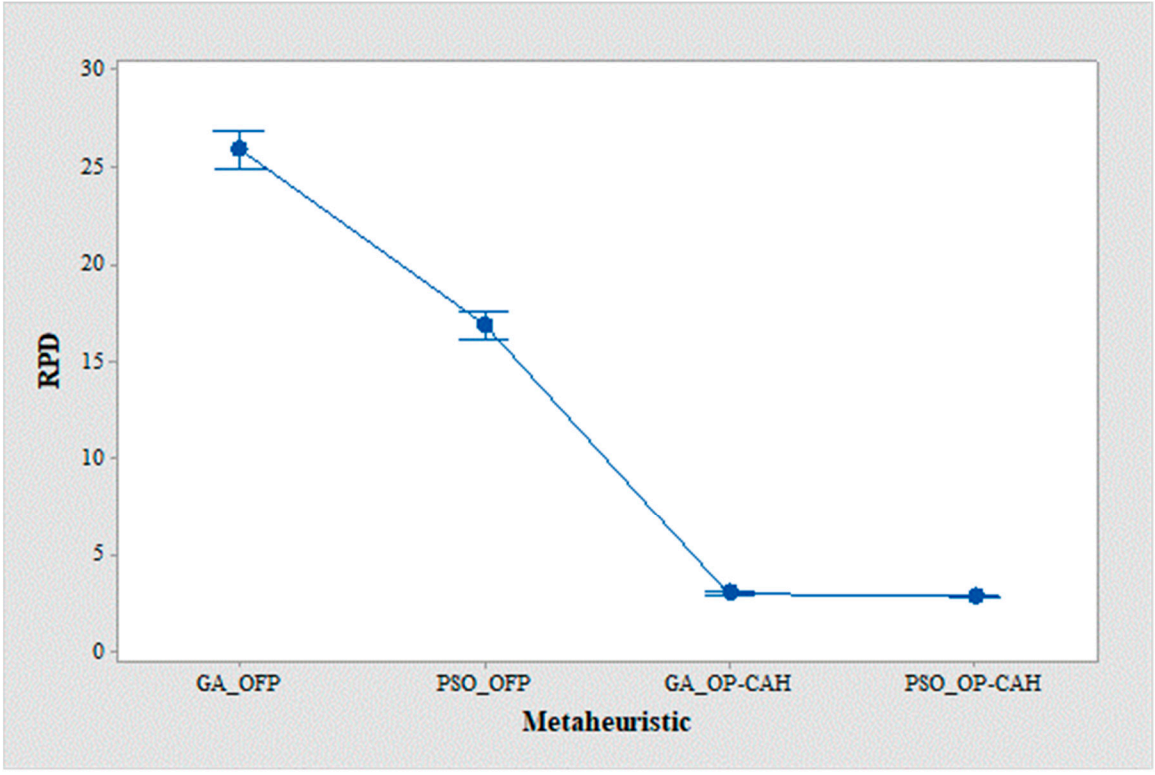
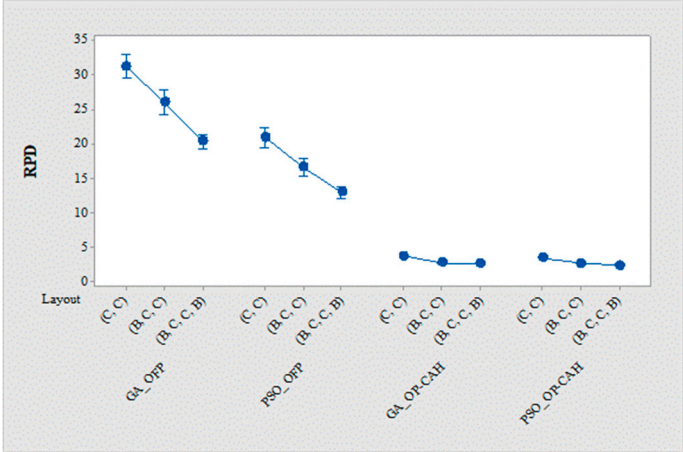
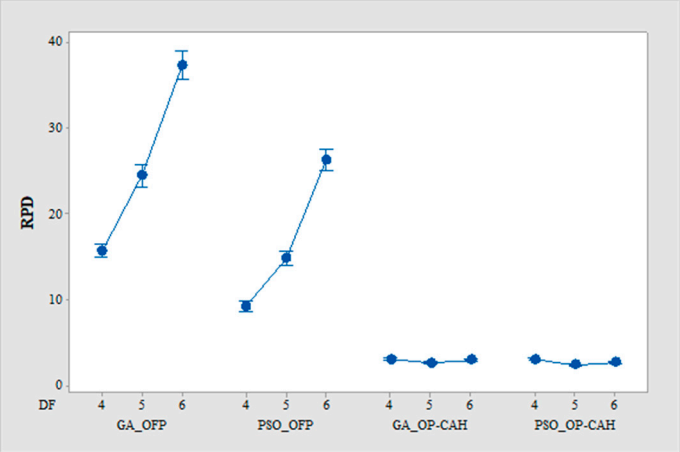


Figure 6. Interval plot of the large-sized experiment with 0.95 confidence level.

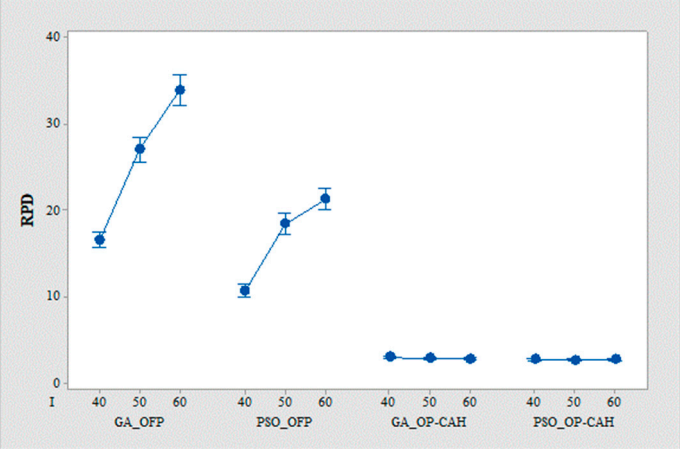
 doi:10.20944/preprints202603.1065.v1



(a) Interval plot for the flowshop layout



(b) Interval plot for the DF



(c) Interval plot for the I

Figure 7. Interval plots for the parameters of the large-sized experiment with 0.95 confidence level.

6. Sensitivity Analysis

We conduct sensitivity analyses to observe the difference in performance gap between *_OFP and *_OP-CAH, and suggest managerial insights for operational strategies to a decision maker. In

 doi:10.20944/preprints202603.1065.v1

this section, we analyze the performance gap depending on the maximum CPU time for identical large-sized instances.

To observe the difference in performance gap between *_OFP and *_OP-CAH, we propose a sensitivity analysis by using the identical instances and the experimental environment in the large-sized experiments. We repeat PSO_OFP and PSO_OP-CAH 10 times for each instance with maximum CPU time as $|DF| \times |I| \times t$ seconds by using a time parameter t. Figure 8 presents an interval plot of the sensitivity analysis by using $ARPD$. PSO_OP-CAH shows the interval plots with no significant difference in the mean value and confidence intervals as the t values increase. In contrast, PSO_OFP shows the interval plots with exponentially decreasing mean value and confidence intervals as increasing the t values increase. Especially, PSO_OFP shows non-overlap with other interval plots and the lowest mean $ARPD$ when the t value exceeds 4.0.

The performance difference in the $ARPD$ between *_OFP and *_OP-CAH is highly related to the maximum CPU time. *_OFP globally searches for the order splitting, assignment, and permutation sequence by the encoded solution, unlike *_OP-CAH. This implies that *_OFP explores a more spacious solution space than *_OP-CAH. This also implies that *_OP-CAH more effectively exploits the solution space within the limited maximum CPU time than *_OFP.

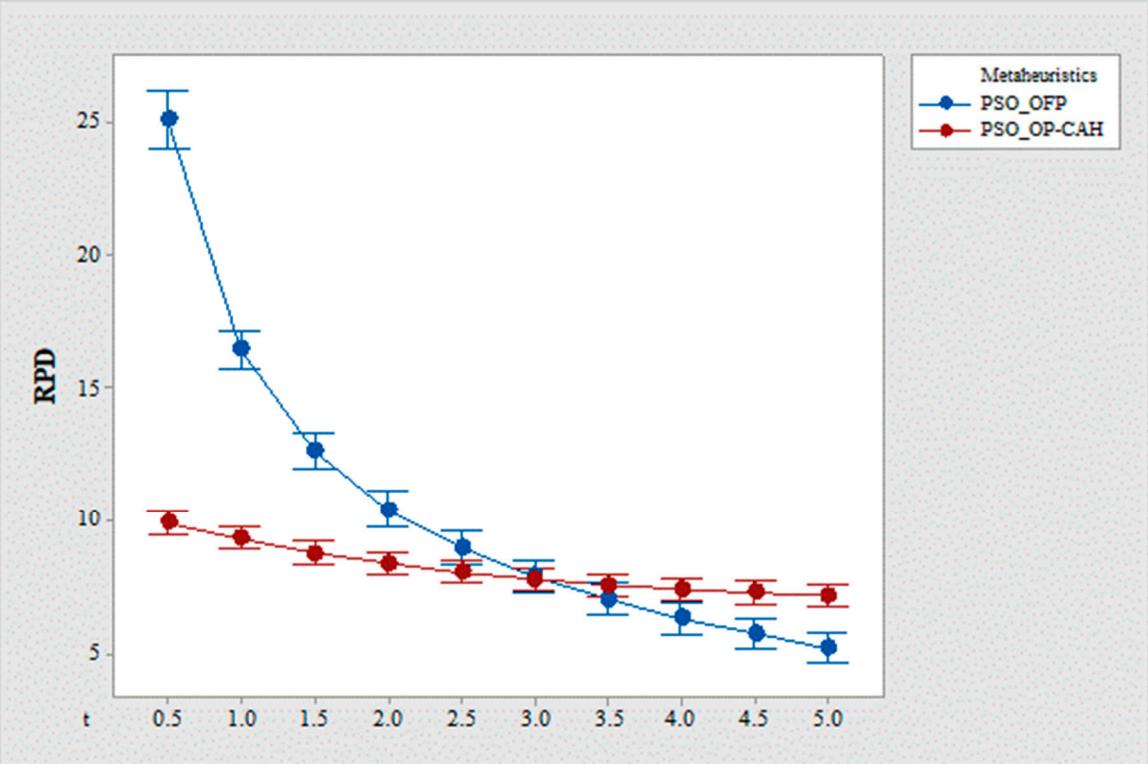


Figure 8. Interval plot of the sensitivity analysis.

We additionally analyze the relationship with the makespan and distributed plant utilization by using the results of the sensitivity analysis in this section. The average plant utilization rate at t (AUR^t) of each distributed plant is calculated as follows:

$$UR_f^t\ (\%) = \frac{\max\limits_{f \in DF} cm_f^t - cm_f^t}{\max\limits_{f \in DF} cm_f^t} \times 100 \tag{41}$$

$$AUR^t\ (\%) = \overline{UR_f^t} \tag{42}$$

, where cm_f^t and UR_f^t mean manufacturing completion time and utilization rate of distributed plant f at time parameter t, respectively. The X-axis means the $ARPD$ value (RPD), and the Y-axis means

 doi:10.20944/preprints202603.1065.v1

the AUR value (AUR). In Figure 9, there are four scatter plots, in which time parameters t are 0.5, 2.0, 3.0, and 5.0, respectively.

For PSO_OP-CAH, as shown in scatter plots (a) – (d) of Figure 9, the cluster becomes denser as the t value increases. However, the cluster does not dynamically move in the direction of smaller RPD and AUR. PSO_OFP forms a cluster with a lower density than PSO_OP-CAH when the t value is 0.5, as shown in the scatter plot (a) of Figure 9. It also forms a denser cluster as the t value increases, and the cluster dynamically moves in the direction of smaller RPD and AUR, as shown in the scatter plots (a)-(d) of Figure 9.

PSO_OP-CAH assigns the sub-order to the plant by focusing on minimizing the completion time of sub-orders via CAH. PSO_OP-CAH shows no noticeable difference in AUR and RPD despite the t-value increasing, unlike PSO_OFP. It implies the relationship between the makespan and plant utilization in the distributed manufacturing environment. Thus, the decision maker has to schedule the distributed plants by considering the plant utilization to minimize the makespan.

From the results of the sensitivity analysis, we propose operational strategies to a decision maker depending upon the decision-making time. If the decision maker has sufficient time for decision-making, we recommend determining the main decision contents of the PS-HF&DS by using the *_OFP to globally search the solution space. However, if the decision maker should decide within a limited time, we recommend using the *_OP-CAH to rapidly converge on a good solution.

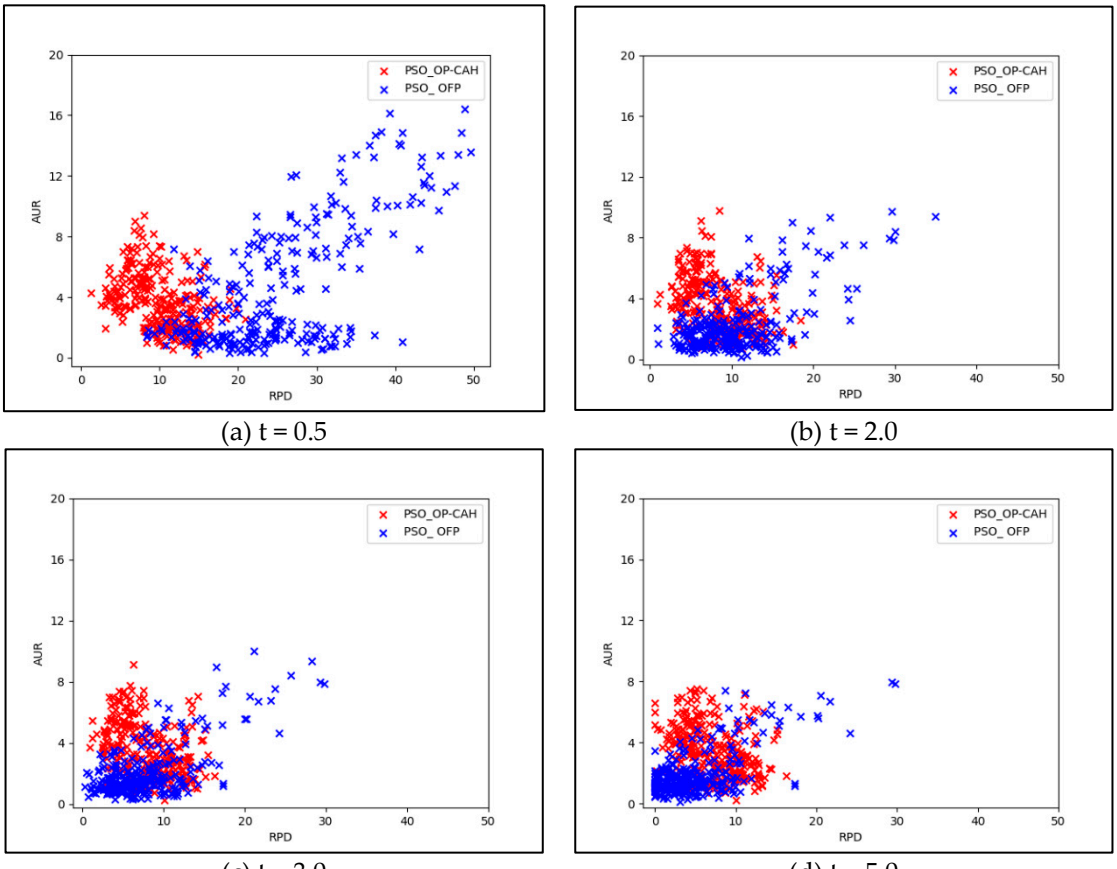


Figure 9. Scatter plot of $\boldsymbol{RPD}$ and $\boldsymbol{AUR}$ with time parameter $\boldsymbol{t}$.

7. Conclusion

In this paper, we address PS-HF&DS considering multiple unrelated plants, a single distinct permutation flowshop-line, hybrid batch-continuous manufacturing, sequence-dependent changeover time, and direct shipping policy within the pharmaceutical supply chain. We formulate the PS-HF&DS as the MILP model and implement *_OFP and *_OP-CAH by applying GA and PSO to effectively and efficiently solve the model. From the numerical experiments, we demonstrate that

 doi:10.20944/preprints202603.1065.v1

*_OP-CAH more effectively finds the near-optimal solutions than *_OFPs within a limited computational time. We additionally conduct the sensitivity analysis of the computational time to analyze the performance gap depending upon the time by comparing PSO_OFP with PSO_OP-CAH. As the computational time increases, *_OFP shows exponentially enhanced performances. Based on the sensitivity analysis result, we propose the operational strategies related to the decision-making time and plant utilization.

Two future research should extend our problem structure. First, we should extend PS-HF&DS with a hybrid flowshop layout with a dynamic order environment. We also should have an extended study of PS-HF&DS related to the routing problem with multiple vehicle-types in the delivery-stage

Data Availability Statement: Data will be made available on request.

Conflicts of Interest: The authors declare that they have no conflict of interest.

Acknowledgements: This work was supported by the Incheon National University Research Grant in 2024 (Grant number: 2024-0185).

Reference

1. Sarkis, M.; Bernardi, A.; Shah, N.; Papathanasiou, M.M. Emerging Challenges and Opportunities in Pharmaceutical Manufacturing and Distribution. *Processes* **2021**, *9*, 457.
2. IQVIA Global Use of Medicines 2023. **2023**.
3. Srai, J.S.; Kumar, M.; Graham, G.; Phillips, W.; Tooze, J.; Ford, S.; Beecher, P.; Raj, B.; Gregory, M.; Tiwari, M.K.; et al. Distributed Manufacturing: Scope, Challenges and Opportunities. *Int. J. Prod. Res.* **2016**, *54*, 6917–6935.
4. Behnamian, J.; Fatemi Ghomi, S.M.T. A Survey of Multi-Factory Scheduling. *J. Intell. Manuf.* **2016**, *27*, 231–249.
5. Perez-Gonzalez, P.; Framinan, J.M. A Review and Classification on Distributed Permutation Flowshop Scheduling Problems. *Eur. J. Oper. Res.* **2023.**
6. Alkhader, W.; Salah, K.; Sleptchenko, A.; Jayaraman, R.; Yaqoob, I.; Omar, M. Blockchain-Based Decentralized Digital Manufacturing and Supply for COVID-19 Medical Devices and Supplies. *IEEE Access* **2021**, *9*, 137923–137940.
7. Ino, E.; Watanabe, K. The Impact of Covid-19 on the Global Supply Chain: A Discussion on Decentralization of the Supply Chain and Ensuring Interoperability. *Journal of Disaster Research* **2021**, *16*, 56–60.
8. Marques, C.M.; Moniz, S.; de Sousa, J.P.; Barbosa-Povoa, A.P.; Reklaitis, G. Decision-Support Challenges in the Chemical-Pharmaceutical Industry: Findings and Future Research Directions. *Comput. Chem. Eng.* 2020, *134*.
9. Inada, Y. Continuous Manufacturing Development in Pharmaceutical and Fine Chemicals Industries. *Mitsui & Co. Global strategic studies institute monthly report* **2019**.
10. Lee, S.L.; O'Connor, T.F.; Yang, X.; Cruz, C.N.; Chatterjee, S.; Madurawe, R.D.; Moore, C.M.V.; Yu, L.X.; Woodcock, J. Modernizing Pharmaceutical Manufacturing: From Batch to Continuous Production. *J. Pharm. Innov.* **2015**, *10*, 191–199.
11. Vieira, M.; Pinto-Varela, T.; Moniz, S.; Barbosa-Póvoa, A.P.; Papageorgiou, L.G. Optimal Planning and Campaign Scheduling of Biopharmaceutical Processes Using a Continuous-Time Formulation. *Comput. Chem. Eng.* **2016**, *91*, 422–444.
12. Nie, Y.; Biegler, L.T.; Wassick, J.M.; Villa, C.M. Extended Discrete-Time Resource Task Network Formulation for the Reactive Scheduling of a Mixed Batch/Continuous Process. *Ind. Eng. Chem. Res.* **2014**, *53*, 17112–17123.
13. Costa, A. Hybrid Genetic Optimization for Solving the Batch-Scheduling Problem in a Pharmaceutical Industry q. *Comput. Ind. Eng.* **2015**, *79*, 130–147.
14. Kopanos, G.M.; Méndez, C.A.; Puigjaner, L. MIP-Based Decomposition Strategies for Large-Scale Scheduling Problems in Multiproduct Multistage Batch Plants : A Benchmark Scheduling Problem of the Pharmaceutical Industry. **2010**, *207*, 644–655.

 doi:10.20944/preprints202603.1065.v1

15. Awad, M.; Mulrennan, K.; Donovan, J.; Macpherson, R.; Tormey, D. A Constraint Programming Model for Makespan Minimisation in Batch Manufacturing Pharmaceutical Facilities. *Comput. Chem. Eng.* **2022**, *156*, 107565.
16. Aguirre, A.M.; Liu, S.; Papageorgiou, L.G. Mixed Integer Linear Programming Based Approaches for Medium-Term Planning and Scheduling in Multiproduct Multistage Continuous Plants. *Ind. Eng. Chem. Res.* **2017**, *56*, 5636–5651.
17. Naderi, B.; Ruiz, R. The Distributed Permutation Flowshop Scheduling Problem. *Comput. Oper. Res.* **2010**, *37*, 754–768.
18. Chen, S.; Pan, Q.K.; Gao, L.; Sang, H. yan A Population-Based Iterated Greedy Algorithm to Minimize Total Flowtime for the Distributed Blocking Flowshop Scheduling Problem. *Eng. Appl. Artif. Intell.* **2021**, *104*, 104375.
19. Karabulut, K.; Kizilay, D.; Tasgetiren, M.F.; Gao, L.; Kandiller, L. An Evolution Strategy Approach for the Distributed Blocking Flowshop Scheduling Problem. *Comput. Ind. Eng.* **2022**, *163*, 107832.
20. Komaki, M.; Malakooti, B. General Variable Neighborhood Search Algorithm to Minimize Makespan of the Distributed No-Wait Flow Shop Scheduling Problem. *Production Engineering* **2017**, *11*, 315–329.
21. Lee, S.J.; Kim, B.S. An Optimization Problem of Distributed Permutation Flowshop Scheduling with an Order Acceptance Strategy in Heterogeneous Plants. *Mathematics* **2025**, *13*.
22. Huang, J.P.; Pan, Q.K.; Miao, Z.H.; Gao, L. Effective Constructive Heuristics and Discrete Bee Colony Optimization for Distributed Flowshop with Setup Times. *Eng. Appl. Artif. Intell.* **2021**, *97*, 104016.
23. Guo, H. wei; Sang, H. yan; Zhang, B.; Meng, L. lei; Liu, L. li An Effective Metaheuristic with a Differential Flight Strategy for the Distributed Permutation Flowshop Scheduling Problem with Sequence-Dependent Setup Times. *Knowl. Based. Syst.* **2022**, *242*, 108328.
24. Meng, T.; Pan, Q.K. A Distributed Heterogeneous Permutation Flowshop Scheduling Problem with Lot-Streaming and Carryover Sequence-Dependent Setup Time. *Swarm Evol. Comput.* **2021**, *60*, 100804..
25. Meng, T.; Pan, Q.K.; Wang, L. A Distributed Permutation Flowshop Scheduling Problem with the Customer Order Constraint. *Knowl. Based. Syst.* **2019**, *184*, 104894.
26. Fu, Y.; Hou, Y.; Chen, Z.; Pu, X.; Gao, K.; Sadollah, A. Modelling and Scheduling Integration of Distributed Production and Distribution Problems via Black Widow Optimization. *Swarm Evol. Comput.* **2022**, *68*, 101015.
27. Hou, Y.; Wang, H.; Fu, Y.; Gao, K.; Zhang, H. Multi-Objective Brain Storm Optimization for Integrated Scheduling of Distributed Flow Shop and Distribution with Maximal Processing Quality and Minimal Total Weighted Earliness and Tardiness. *Comput. Ind. Eng.* **2023**, *179*, 109217.
28. Feng, X.; Zhao, F.; Jiang, G.; Tao, T.; Mei, X. A Tabu Memory Based Iterated Greedy Algorithm for the Distributed Heterogeneous Permutation Flowshop Scheduling Problem with the Total Tardiness Criterion. *Expert Syst. Appl.* **2023**, 121790.
29. Fernandez-Viagas, V.; Molina-Pariente, J.M.; Framinan, J.M. New Efficient Constructive Heuristics for the Hybrid Flowshop to Minimise Makespan: A Computational Evaluation of Heuristics. *Expert Syst. Appl.* **2018**, *114*, 345–356.
30. Fernandez-Viagas, V.; Perez-Gonzalez, P.; Framinan, J.M. Efficiency of the Solution Representations for the Hybrid Flow Shop Scheduling Problem with Makespan Objective. *Comput. Oper. Res.* **2019**, *109*, 77–88.
31. Hatami, S.; Ruiz, R.; Andrés-Romano, C. The Distributed Assembly Permutation Flowshop Scheduling Problem. *Int. J. Prod. Res.* **2013**, *51*, 5292–5308.
32. Yang, S.; Xu, Z. The Distributed Assembly Permutation Flowshop Scheduling Problem with Flexible Assembly and Batch Delivery. *Int. J. Prod. Res.* **2021**, *59*, 4053–4071.
33. Ying, K.-C.; Pourhejazy, P.; Cheng, C.-Y.; Syu, R.-S. Supply Chain-Oriented Permutation Flowshop Scheduling Considering Flexible Assembly and Setup Times. *Int. J. Prod. Res.* **2023**, *61*, 258–281.
34. Mraihi, T.; Driss, O.B.; El-Haouzi, H.B. Distributed Permutation Flow Shop Scheduling Problem with Worker Flexibility: Review, Trends and Model Proposition. *Expert Syst. Appl.* **2023**, 121947.
35. Aguirre, A.M.; Liu, S.; Papageorgiou, L.G. Mixed Integer Linear Programming Based Approaches for Medium-Term Planning and Scheduling in Multiproduct Multistage Continuous Plants. *Ind. Eng. Chem. Res.* **2017**, *56*, 5636–5651.

 doi:10.20944/preprints202603.1065.v1

36. Gao, J.; Chen, R. A Hybrid Genetic Algorithm for the Distributed Permutation Flowshop Scheduling Problem. *International Journal of Computational Intelligence Systems* **2011**, *4*, 497–508.
37. Li, X.; Zhang, X.; Yin, M.; Wang, J. A Genetic Algorithm for the Distributed Assembly Permutation Flowshop Scheduling Problem. *2015 IEEE Congress on Evolutionary Computation, CEC 2015 - Proceedings* **2015**, 3096–3101.
38. Allali, K.; Aqil, S.; Belabid, J. Distributed No-Wait Flow Shop Problem with Sequence Dependent Setup Time: Optimization of Makespan and Maximum Tardiness. *Simul. Model. Pract. Theory* **2022**, *116*, 102455.
39. Liu, H.; Gao, L.; Pan, Q. A Hybrid Particle Swarm Optimization with Estimation of Distribution Algorithm for Solving Permutation Flowshop Scheduling Problem. *Expert Syst. Appl.* **2011**, *38*, 4348–4360.
40. Marichelvam, M.K.; Geetha, M.; Tosun, Ö. An Improved Particle Swarm Optimization Algorithm to Solve Hybrid Flowshop Scheduling Problems with the Effect of Human Factors – A Case Study. *Comput. Oper. Res.* **2020**, *114*, 104812.
41. Pan, Q.-K.; Tasgetiren, M.F.; Liang, Y.-C. A Discrete Particle Swarm Optimization Algorithm for the No-Wait Flowshop Scheduling Problem. *Comput. Oper. Res.* **2008**, *35*, 2807–2839.
42. Vallada, E.; Ruiz, R. A Genetic Algorithm for the Unrelated Parallel Machine Scheduling Problem with Sequence Dependent Setup Times. *Eur. J. Oper. Res.* **2011**, *211*, 612–622.
43. Kim, Y.J.; Jang, J.W.; Kim, D.S.; Kim, B.S. Batch Loading and Scheduling Problem with Processing Time Deterioration and Rate-Modifying Activities. *Int. J. Prod. Res.* **2022**, *60*, 1600–1620.