

Integrated Model for Pickup, Production, and Batch Delivery for a Single Item with Multiple Due Dates to Minimize Total Relevant Cost

Murni Dwi Astuti^a, Sukoyo^{a,*}, Andi Cakravastia^a, and Abdul Hakim Halim^a

^aFaculty of Industrial Technology, Institut Teknologi Bandung, Jl. Ganesha No. 10, Bandung 40132, Indonesia

*Corresponding author. E-mail: sukoyo@itb.ac.id

ABSTRACT

This study integrates pickup, production, and batch delivery activities within a make-to-order production system. It distinguishes between holding costs for parts, work-in-process, and finished goods borne by either the manufacturer or the customer to meet the demands of a single customer with varying due dates. The problem is formulated as a Mixed-Integer Nonlinear Programming model that simultaneously determines the number of batches, batch sizes, and schedules for all three activities; a backward scheduling approach is employed to minimize total relevant costs based on the Actual Flow Time (AFT) metric. Decisions regarding batch quantity, size, and production scheduling both influence and are influenced by decisions concerning delivery and pickup activities. Due to the complexity of the model and the inclusion of continuous, binary, and integer decision variables, an analytical solution is not feasible; consequently, an exact method using the Gurobi solver was applied. With a computation time limit of 10.800 seconds, Gurobi failed to find an optimal solution, identifying only the best feasible solution for the five tested datasets; therefore, a Genetic Algorithm (GA) was developed. The proposed GA is capable of solving the problem and generating solutions that approximate the best feasible solutions found by the exact method, while requiring significantly less computation time.

KEYWORDS: Pickup; Production; Delivery; Actual Flow Time; Backward Scheduling.

1. Introduction

For most businesses, providing superior customer service cost-effectively is a key strategic objective [1]. However, manufacturing firms must not only capture market share but also optimize operations and flexibility to enhance profitability [2]. Consequently, in addition to integrating various decision-making processes, manufacturers are encouraged to collaborate with customers and suppliers [3]. This perspective aligns with the notion that business systems have undergone significant changes in recent years: business systems are no longer characterized by competition between individual businesses, but rather by cooperation within integrated enterprise systems [4].

To ensure efficiency and achieve a competitive advantage in the business world, cross-functional integration and coordination within an integrated corporate system are crucial. Traditional optimization methods limited to a single function such as procurement, production, or distribution may fail to yield an optimal solution when viewed from a holistic system perspective [5]. Effective coordination among these functions has been shown to reduce total costs, minimize inventory, enhance customer service levels, and accelerate response times to market demand [6]. Such cross-functional coordination operates through integration mechanisms.

Research on production-delivery integration has expanded rapidly over the past two decades. Berghman et al. [7] reviewed 65 studies published since 2010 addressing the integration of production scheduling and delivery vehicle routing; their analysis indicates that an integrated approach significantly improves overall performance yielding average gains of 5% to 20% compared to sequential solutions by reducing waiting times and inventory levels. Prasetyaningsih et al. [8] developed an integration of batch production and delivery scheduling model with multiple due dates using a backward scheduling approach, incorporating step-wise inventory costs to minimize total costs. Similarly, Agnetis et al. [9] constructed a batch production and delivery scheduling model with fixed departure times that accounts for inventory holding costs. Karimi & Davoudpur [10] extended the scope of research to the supply chain level by integrating production and delivery decisions within a multi-plant system while accommodating step-wise inventory cost structures. Furthermore, Yin et al. [11] formulated an integrated model encompassing due-date setting, production, inventory, and batch delivery for multiple agents with limited capacity; the model employs a bi-criterion analysis to optimize the costs of two competing agents, alongside an evaluation of computational complexity and algorithmic solutions. Han et al. [12] complemented this body of research by integrating

production, inventory, and outbound distribution operations into a single modelling framework featuring fixed departure times. Beyond cost and time considerations, Yagmur & Kesen [13] introduce a sustainability dimension to integrated production and distribution scheduling, accounting for early/late delivery costs alongside the energy consumption of machines and vehicles.

Research on the integration of production and delivery specifically considers delivery routing; for instance, Azad et al. [14] optimize integrated production scheduling and vehicle routing problems involving batch deliveries to multiple customers. Marandi & Fatemi Ghomi [15] address the practical modelling of integrated production and distribution scheduling across multiple plants to minimize total delay and transportation costs by optimizing limited-capacity fleet routing within a plant network configuration. Mohammadi et al. [16] examine the integration of bi-objective production scheduling and delivery routing in make-to-order (MTO) supply chains to maximize customer satisfaction without significantly increasing operational costs; this study was the first to integrate flexible job shop scheduling with vehicle routing decisions involving time windows (VRPTW). Kuhn et al. [17] address the integrated problem of order batching and vehicle routing for micro-grocery stores, which are characterized by small order volumes and frequent deliveries. Hrabec et al. [18] analyze the value of integrated planning for production, inventory, and routing decisions, finding that this integration yields an average cost saving of 11, 08% compared to sequential planning.

Production and delivery integration is also implemented for perishable products; for instance, Devapriya et al. [19] address the integration of production and distribution scheduling to minimize costs for perishable goods, accounting for constraints such as limited product shelf life, demand fulfilment obligations within the planning horizon, and the use of multiple trucks to ensure delivery before expiration. Farahani et al. [20] integrate short-term production planning and the distribution of perishable catering food using an iterative scheme designed to shorten delivery lead times and maintain product quality. Lacomme et al. [21] formulate an integrated multi-vehicle production and transportation scheduling model for a single perishable product, proposing a GRASP $\times$ ELS algorithm that has been shown to yield significantly superior solution efficiency and reduce average makespan compared to previous methods.

These studies demonstrate that integrating production and delivery decisions can yield significant cost savings compared to sequential approaches. However, they often assume the immediate availability of raw materials or parts required for production, thereby overlooking the complexities and costs associated with sourcing and transporting materials from suppliers. Recognizing that neglecting the upstream (supplier) side can lead to inefficiencies, recent studies have begun to expand system boundaries by formulating integrated procurement, production, and distribution (PPD) strategies. Maheswari et al. [22] examined PPD integration in a medium-scale food processing company to minimize makespan, using production quantities and batch sizes as parameters while excluding inventory holding costs. Meanwhile, Rosyidi et al. [23] developed a model that simultaneously integrates decisions regarding raw material procurement, lot sizing, transport fleet selection, and production scheduling to

maximize total profit; this model accounts for quantity discount schemes, raw material defect rates, backlogs, and production time capacity constraints. Khanra et al. [24] developed a model integrating production, inventory, and delivery policies involving a single supplier, manufacturer, and retailer to maximize average total profit; their approach utilized variable sequential shipment sizes, determined supplier production and downtime levels, and analyzed idle costs across all entities using a forward scheduling method.

Although literature on PPD integration has expanded rapidly, significant limitations and critical research gaps remain regarding cost representation and operational-level details. Benbouja et al. [25] developed an integrated PPD model to align procurement, production, and distribution decisions within a mass-customization supply chain; however, this model operates at an aggregate, period-based tactical planning level, failing to account for detailed schedules or real-time sequencing of individual operational activities on the shop floor. Conversely, studies at the operational level often overlook crucial material transfer cost components; for instance, Nikandish et al. [26] developed an integrated optimization model to align raw material procurement, production, and delivery decisions using general cycle-time assumptions, yet completely disregarded transportation costs for both raw material procurement and product delivery. Furthermore, even comprehensive operational models such as the one proposed by Tibaldo et al. [27], which addresses the simultaneous integration of batching (or lot sizing), production scheduling, and product delivery, assume that raw materials are ready when production begins, thereby ignoring the dynamics of material pickup from suppliers.

Based on a literature review, existing models still view this problem from a partial perspective, whether by ignoring the dynamics of raw material availability from suppliers, operating at an aggregate level, or failing to consider cost elements associated with the integrated scheduling of raw material procurement, production, and batch delivery. Furthermore, existing integration models generally treat inventory levels as variables determined by their impact on total costs. Therefore, this study holistically integrates component procurement, production scheduling, and batch delivery using a backward scheduling approach that accounts for holding costs. Holding costs are differentiated by category: parts, work-in-process, finished goods at the manufacturer, and finished goods at the customer's site. These holding cost calculations are based on the concept of actual flow time, wherein parts need not be on the production floor at the time $t = 0$ but are required only when needed for production. Inventory levels at each stage emerge from the integration of procurement, production, and delivery activities, thereby yielding an optimal solution that balances the trade-offs inherent in such integration.

2. Mathematical Model

This section describes the problem and the assumptions used in developing the model, as well as the sets and indices, parameters, and decision variables employed in the model.

2.1. Notation

Table 1 presents the notation used in this research model. The notation is categorized into sets and indices, parameters, and decision variables.

Table 1. Notation used in the proposed model

Notation	Definition
Set and Indices	
A	Set of pickup vehicles, $\{A = K_1, K_2, \dots, A \}$ indexed by a
H	Set of delivery batches, $\{H = 1, 2, \dots, H \}$, indexed by h
I	Set of production batches, $\{I = 1, 2, \dots, I \}$, indexed by i
Ω	Set of nodes, $\{\Omega = N_0, N_1, N_2, \dots, N_b\}$, where N_0 is a manufacturer node, indexed by f, g
Φ	Set of suppliers, $\Phi = \Omega \setminus N_0$, indexed by f, g
R	Set of customer orders, $\{R = 1, 2, \dots, R \}$ indexed by r
Parameters	
c	Capacity of rack (unit product/ rack)
c^c	Unit cost of rack procurement (\$/ unit)
c^k	Cost of pickup (\$/ km)
c^m	Product holding cost at customer (\$/ unit product/ period)
c^r	Product holding cost at manufacturer (\$/ unit product/ period)
c^s	Part holding cost at manufacturer (\$/ unit part/ period)
c^t	Cost of delivery (\$/ delivery)
c^w	Work in process (WIP) holding cost at the manufacturer (\$/ unit product/ period)
$d_{(r)}$	Due date requested to r
$Jr_{(f,g)}$	Inter-node distance (km)
k	Vehicle capacity for delivery (batch /vehicle)
M	Sufficient big number
n	Total demand for product (units of product)
v	Duration of Delivery (hours/ delivery)
$Q_{(r)}$	The quantity of product demand volume that must be fulfilled in the due date period r
p	Part loading time (hour/supplier)
s	Set-up time/batch
t	Time of product processing (hours/product)
Vl	Pickup vehicle speed (Km/hours)
W	Vehicle capacity for pickup (part/ vehicle)
Continuous Variables	
$Bp_{(i)}$	Start time of production batch i
$BS_{(i)}$	Start time of the setup batch i
$Cp_{(i)}$	Production completion time for the batch i
$Bd_{2(i)}$	Start time of delivery batch i
$Cd_{(i)}$	Arrival time of the delivery batch i
$Bd_{(h)}$	Start time of Delivery h
$Sp_{(h)}$	Return time of Delivery h to the manufacturer
$L_{(f,a)}$	Departure time of the vehicle a from the supplier f
$Wk_{(a)}$	Arrival time of the pickup vehicle a at the manufacturer
$\delta_{(i,f,a)}$	An auxiliary variable is the arrival time of raw materials from the supplier f via vehicle a for batch i
Integer variables	
$q_{(i,r)}$	The quantity of product volume from the batch i allocated to meet the target due date period r
$Q_{(i)}$	Production quantity of the batch i
$Z_{(f,a)}$	The load of vehicle a when leaving the supplier f
Binary variables	
$v_{(i,r)}$	Binary variable, 1 if production batch i is allocated to meet demand on due date r , 0 otherwise
$O_{(i)}$	Binary variable, equals 1 if batch i is active, and 0 otherwise
$Jp_{(h)}$	Binary variable, equals 1 if Delivery to h is active, and 0 otherwise
$Xp_{(h,i)}$	Binary variable, equals 1 if batch i is assigned to Delivery h , and 0 otherwise
$X_{(f,g,a)}$	Binary variable, equals 1 if inbound vehicle a travels from the location f to g , and 0 otherwise
$Y_{(a)}$	Binary variable, equals 1 if the vehicle a is used for part pickup, and 0 otherwise

Notation	Definition
$\Theta_{(f,a)}$	Binary variable, equals 1 if the vehicle a schedules a visit to the supplier f on its route, and 0 if it skips that supplier
$U_{(i,f,a)}$	Binary variable, 1 if the load of raw materials at the supplier f picked up by vehicle a is intended for a production batch i ; 0 otherwise
AFT Calculation	
$F_{(i,f,r)}$	AFT one part in one flow
$F_{(i,f,r)}^l$	AFT all parts in one flow
$F_{(i)}^l$	Total AFT parts in one batch
F^l	AFT total parts in the entire batch

2.2. Problem Description

This study develops the models discussed in Prasetyaningsih et al. [8] and Astuti et al. [28]. The extension of the model in [8] incorporates part pickup from suppliers and allows a single production batch to satisfy multiple order due dates, while the extension of the model in [28] addresses non-simultaneous part arrivals and customer demand involving multiple due dates. Figure 1 illustrates the problem addressed in this study. A single manufacturer receives multiple customer orders, each with a specific due date and a corresponding quantity, $Q_{(r)}$ (where $r = 1, \dots, R$). Products are manufactured using a batch production system. Orders are divided into production batches i , with each batch having a size of $Q_{(i)}$. Before processing begins, the machine undergoes a setup lasting s time units. Processing for a batch starts at time $Bp_{(i)}$ taking $t * Q_{(i)}$ time and stops at time $Cp_{(i)}$.

During processing, products in a specific batch i must wait until the entire batch has finished processing, incurring a holding cost of c^w per product per period. Once processed, products from each batch are shipped to the customer at time $Bd_{2(i)}$ and arrive at time $Cd_{(i)}$; the shipping system utilizes a single vehicle with a capacity of k batches, resulting in a waiting period between processing completion and shipment that incurs a holding cost of c^r per product per period. Since customer demand involves multiple due dates and a single production batch can satisfy demand associated with different due dates, batches allocated to meet demands with varying due dates are shipped according to the earliest due date; this creates a waiting period between the product's arrival at the customer and its specific due date, incurring a holding cost of c^m per product per period. Each production batch shipped to the customer utilizes racks costing c^c per rack, with a shipping cost of c^t per shipment. The product is manufactured with parts from several suppliers, and each part type has only one supplier. The parts are delivered to the collection points by a limited-capacity vehicle in a "milk-run" system, and the collection cost is c^k per km. Each product needs all parts from all suppliers; that is, the parts must be in the manufacturing plant before its starting time of batch production. Not all parts from the same supplier need to be received at the same time, because parts awaiting use in production incur a holding cost c^s Per part unit per period. Upon arrival at the manufacturer, parts are sorted based on production batch size and quantity, then loaded onto racks with a capacity matching that of the machines. These racks are subsequently used to transport the production batches to the customer.

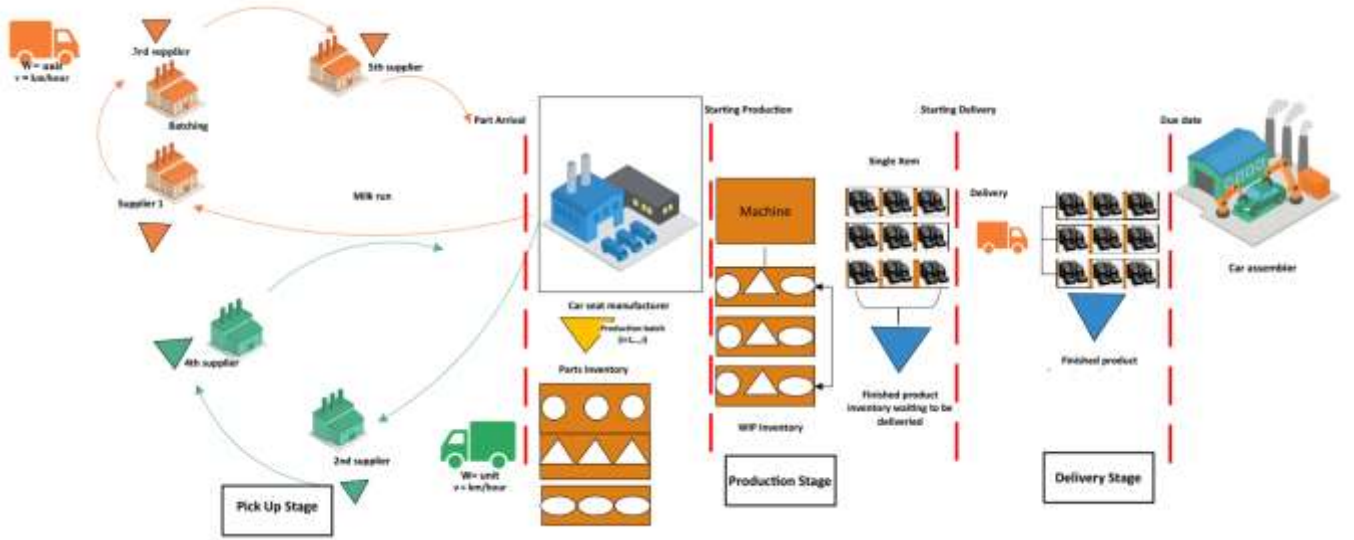


Figure 1. Problem Description

This analysis indicates that solving these problems individually can be inefficient for customer delivery, production scheduling, batch sizing, and pickup. These inefficiencies can lead to expensive Work-In-Process (WIP) inventory, poor vehicle utilization, or missed on-time customer deliveries. Therefore, the integrated model proposed in this study addresses these challenges by combining the three activities pickup, production, and batch delivery to minimize total relevant costs. By clearly distinguishing holding costs at each stage from parts and WIP to finished goods awaiting shipment and finished goods arriving early at the customer's site, the model enables an optimal solution that balances pickup, holding, and delivery costs, which means minimizing total relevant costs while prioritizing adherence to customer due dates.

2.3. Actual Flow Time of Pickup, Production, and Batch Delivery

The Actual Flow Time (AFT) performance measure for batch production systems was proposed by Halim et al. [29] and is defined as the duration a batch spends on the production floor from the start of production until its due date. Prasetyaningsih et al. [8] developed the AFT concept to integrate production and batch delivery, calculating AFT from the time a job enters the production floor until its due date. In this study, the integration is extended by considering part pickups from various suppliers, such that the AFT is calculated from the moment the parts arrive at the manufacturing site until the due date. Furthermore, a single production batch can be used to meet demands with different due dates. Consequently, the flow of parts is not serial, as a single product requires one unit of a part from each supplier, a production batch can contain multiple products, and a production batch can satisfy demands with varying due dates. Therefore, the AFT formulation is developed in stages, encompassing:

- Defining the time span for a single part flow in Equation (1) and all parts in a flow in Equation (2)
- Breaking down that span into five sequential time stages, in Equation (3) as summarized in Table 2
- Recombining the five stages with varying weights for a single production batch, in Equation (4)

- Summing the results for the entire system and formulating the objective function, in equations (6) through (7)

The model proposed by Prasetyaningsih et al. [8] assumes that all parts are available on the production floor exactly when needed, thereby disregarding the concepts of part arrival times and holding costs. In this model, part arrival times are variables determined by the pickup vehicle's routing, and a single production batch can be used to fulfill orders with several due dates. Consequently, the AFT for a production batch cannot be expressed as a single value because the input is not singular, as parts originate from multiple suppliers, and the output is not singular either, since the batch's contents can satisfy multiple due dates. Therefore, the breakdown of AFT begins with a smaller unit known as a "flow." A flow is defined as a set of parts from a single supplier f , intended for a production batch i that fulfills a due date order r with a quantity $q_{(i,r)}$; thus, the flow is characterized by three indices (i, f, r) . All parts within a single flow arrive at the manufacturer simultaneously since they originate from the same supplier and are transported by the same vehicle and share the same due date because they are allocated to the same order; consequently, all parts in the flow possess an identical AFT value. Breaking the flow down further, for instance, by calculating the AFT for each part individually yields no additional information, as the result remains the same for all parts within the same flow. Thus, the flow is selected as the unit of breakdown because it is sufficiently granular compared to the batch yet requires no further subdivision due to the uniformity of its contents; further splitting would merely repeat the same numerical value.

Let us assume that batch i receives parts from supplier f via vehicle a , and that order r is one of the orders served by that batch. The AFT for a single part within flow (i, f, r) is expressed as:

$$F_{(i,f,r)} = d_{(r)} - \sum_{a \in A} \delta_{(i,f,a)}, \forall i \in I, \forall f \in \Phi, \forall r \in R \mid v_{(i,r)} = 1 \quad (1)$$

The summation over index a in Equation (1) acts as a selector rather than an accumulator; consequently, the equation holds true whether all suppliers are served by the same vehicle or by different vehicles. This is ensured by Equation (19) and (39). Equation (19) stipulates that for every pair consisting of batch i and supplier f , the pair is assigned to exactly one vehicle if the batch is active. Equation (39) establishes that if a vehicle is not assigned to perform the pickup, then $\delta_{(i,f,a)} = 0$; conversely, if a vehicle is assigned, it yields $\delta_{(i,f,a)} = Wk_{(a)}$ (since $u_{(i,f,a)} = 1$). Thus, the summation across all vehicles contains only one non-zero term: that corresponding to the return of the specific vehicle that actually served the pair. As a result, the arrival time is determined by the vehicle rather than the supplier; therefore, two suppliers picked up by the same vehicle share an identical arrival time, whereas two suppliers picked up by different vehicles will have different arrival times.

The interval in Equation (1) ends at the due date $d_{(r)}$ rather than at the arrival time of the product $Cd_{(i)}$, because products arriving early incur holding costs at the customer location until their due date. Since all parts in a single flow share the same AFT, the total AFT for all parts in that flow is obtained by multiplying the AFT by the quantity of parts, $q_{(i,r)}$:

$$F_{(i,f,r)}^l = q_{(i,r)} \left(d_{(r)} - \sum_{a \in A} \delta_{(i,f,a)} \right), \quad \forall i \in I, \forall f \in \Phi, \forall r \in R \mid v_{(i,r)} = 1 \quad (2)$$

Equation (2) applies only to a single supplier and a single order; the AFT for the entire batch cannot be obtained by simply summing Equation (2) for the two, because the number of waiting units changes according to the material's state, as shown in Figure 2.

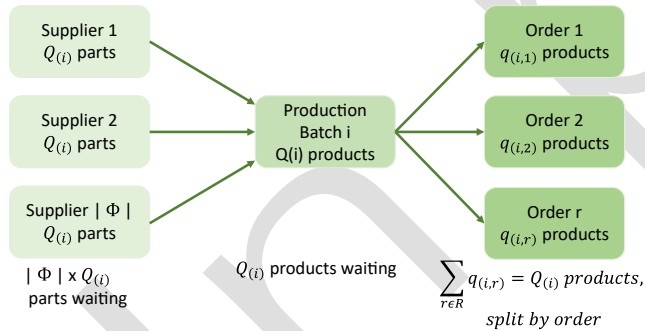


Figure 2. Changes in the number of units waiting for each stage

Before production begins, the manufacturer holds parts from $|\Phi|$ suppliers, each supplying $Q_{(i)}$ units, resulting in a total of $|\Phi| \times Q_{(i)}$ parts awaiting use in production, which incurs holding costs. Since each product unit requires parts from all suppliers, Equation (40) mandates that all parts must arrive before batch processing commences. If the parts for a batch are collected by multiple vehicles, earlier-arriving pickups face longer wait times; conversely, if all suppliers are served by a single route, all shipments arrive simultaneously, resulting in equal wait times. Thus, wait times are calculated for each supplier by accounting for both scenarios. This applies only until production begins and is not used in subsequent stages. Once production starts, parts from all suppliers have been transformed into $Q_{(i)}$ product units, and it is this quantity that

incurs holding costs in the ensuing stages. After the products reach the customer, the $Q_{(i)}$ units are sorted according to the orders they fulfill, each consisting of $q_{(i,r)}$ units, meaning the quantity of items awaiting processing varies at each stage; consequently, aggregation must be performed stage by stage. The interval in Equation (1) is first divided into five consecutive segments:

$$d_{(r)} - \sum_{a \in A} \delta_{(i,f,a)} = \left(Bp_{(i)} - \sum_{a \in A} \delta_{(i,f,a)} \right) + (Cp_{(i)} - Bp_{(i)}) + (Bd2_{(i)} - Cp_{(i)}) + (Cd_{(i)} - Bd2_{(i)}) + (d_{(r)} - Cd_{(i)}), \quad \forall i \in I, \forall f \in \Phi, \forall r \in R \mid v_{(i,r)} = 1 \quad (3)$$

Equation (3) is an algebraic identity, an equality that holds true regardless of the values substituted, rather than a constraint or a new assumption, as the middle terms on the right-hand side cancel each other out, leaving exactly the left-hand side. This equation accounts for the duration of each stage; Table 2 summarizes the number of items waiting at each stage, along with the associated multipliers and cost rates.

Table 2. Material flow at each stage

No	Duration	Material status	Quantity	Cost
1	$\delta_{(i,f,a)} \rightarrow Bp_{(i)}$	Part at the manufacturer	$Q_{(i)}$, summed up for all suppliers	c^s
2	$Bp_{(i)} \rightarrow Cp_{(i)}$	Work in process	$Q_{(i)}$	c^w
3	$Cp_{(i)} \rightarrow Bd2_{(i)}$	Finished products at the manufacturer	$Q_{(i)}$	c^r
4	$Bd2_{(i)} \rightarrow Cd_{(i)}$	Finished products in transit	$Q_{(i)}$	Fix/delivery
5	$Cd_{(i)} \rightarrow d_{(r)}$	Finished products at the customer	$q_{(i,r)}$, summed up for all due dates	c^m

Given the data in Table 2, with that multiplier, the actual flow time of all units in the production batch i is formulated as:

$$F_{(i)}^l = \sum_{f \in \Phi} Q_{(i)} \left(Bp_{(i)} - \sum_{a \in A} \delta_{(i,f,a)} \right) + Q_{(i)} [(Cp_{(i)} - Bp_{(i)}) + (Bd2_{(i)} - Cp_{(i)}) + (Cd_{(i)} - Bd2_{(i)})] + \sum_{r \in R} q_{(i,r)} (d_{(r)} - Cd_{(i)}), \quad \forall i \in I \quad (4)$$

Equation (4) is not obtained by summing Equation (2) over all suppliers and orders. Such a summation would count the second through fifth terms $|\Phi|$ times since these components are summed for each supplier whereas, once production begins, there are only $Q_{(i)}$ units of product, not $|\Phi| \times Q_{(i)}$. In the fifth segment, the multiplier changes from $Q_{(i)}$ to $q_{(i,r)}$ because the batch contents have been allocated to specific orders. This transition is valid as it is guaranteed by two mass balance equations: Equation (8), which ensures the entire batch content is fully distributed among the orders ($\sum_{i \in I} q_{(i,r)} = Q_{(r)}, \forall r \in R$).

R_i) and Equation (10), which ensures each order's demand is met ($\sum_{r \in R} q_{(i,r)} = Q_{(i)}, \forall i \in I$).

Based on Equation (24), the length of the second segment is proportional to the batch size; consequently, its product with the multiplier $Q_{(i)}$, yields a quadratic form, classifying this model as an integer non-linear programming problem, and the second segment becomes:

$$Q_{(i)}(Cp_{(i)} - Bp_{(i)}) = Q_{(i)}tQ_{(i)}, \forall i \in I \quad (5)$$

Because processing relies on a batch production system, products completed early in a batch must wait until the final unit finishes processing; consequently, the total waiting time for work-in-process increases quadratically with batch size, creating pressure to use smaller batches. Conversely, delivery costs (c^t) and rack procurement costs (c^c) incurred per delivery and per active batch drive the system toward larger batch sizes. By applying Equation (5) to Equation (4) and summing the results across all batches, the total actual flow time is obtained:

$$F^l = \sum_{i \in I} \sum_{f \in \Phi} Q_{(i)} \left(Bp_{(i)} - \sum_{a \in A} \delta_{(i,f,a)} \right) + \sum_{i \in I} t Q_{(i)}^2 \quad (6)$$

$$+ \sum_{i \in I} Q_{(i)} (Bd_{2(i)} - Cp_{(i)})$$

$$+ \sum_{i \in I} Q_{(i)} v$$

$$+ \sum_{i \in I} \sum_{r \in R} q_{(i,r)} (d_{(r)} - Cd_{(i)})$$

The fourth term of Equation (6) is evaluated using Equation (9). Based on Equation (9), the fourth segment evaluates to $\sum_{i \in I} Q_{(i)} v = n * v$, a constant that does not involve decision variables and can therefore be removed without altering the optimal solution.

The following is a small numerical example illustrating how AFT is calculated for one batch with two suppliers and two due dates:

Suppose production batch i has a size of $Q_{(i)} = 20$ units ($t = 0,5; v = 20$) and is used to fulfill two orders: ($d_{(1)} = 90, q_{(i,r)} = 13$) and ($d_{(2)} = 100, q_{(i,2)} = 7$). Since AFT employs a backward scheduling approach, the departure time of the last delivery plus the delivery duration must exactly match the earliest due date; thus, the departure time for the last delivery is $90 - 20 = 70$. Only a single production batch is processed, requiring just one shipment; consequently, the processing completion time ($Cp_{(i)}$) coincides exactly with the delivery departure time, and the batch processing start time becomes: $Cp_{(i)} - tQ_{(i)} = 70 - (0,5 * 20) = 60$.

To produce this batch, parts are required from two suppliers, collected using two different vehicles with $\delta_{(1,1,1)} = 40$ and $\delta_{(1,2,2)} = 60$.

The AFT calculation is performed using Equation (6), but term 4 is excluded; this is because, when calculating total costs, shipping is based on delivery frequency rather than duration, making the AFT for this example:

- AFT for parts awaiting processing at the manufacturer:

$$\sum_{i \in I} \sum_{f \in \Phi} Q_{(i)} \left(Bp_{(i)} - \sum_{a \in A} \delta_{(i,f,a)} \right)$$

$$= 20 (60 - 40) + 20 (60 - 60)$$

$$= 400$$

- AFT for WIP:

$$\sum_{i \in I} t Q_{(i)}^2 = 0,5 * (20) * (20) = 200$$

- AFT for products awaiting delivery:

$$\sum_{i \in I} Q_{(i)} (Bd_{2(i)} - Cp_{(i)}) = 20 (70 - 70) = 0$$

- AFT for products waiting at the customer:

$$\sum_{i \in I} \sum_{r \in R} q_{(i,r)} (d_{(r)} - Cd_{(i)})$$

$$= 13 (90 - 90) + 7 (100 - 90)$$

$$= 70$$

2.4. Mathematical Model Formulation

The assumptions adopted in this study are:

1. All customer orders with due dates $d_{(r)}$ are known at the beginning of the scheduling period.
2. Suppliers are assumed to produce a unique part type that meets specified manufacturer requirements, with the requirement for each part type per finished product unit being identical across all.
3. There are no part holding costs at the supplier's location or during transit to the manufacturer.
4. There is no unloading activity at the manufacturer, so the vehicle's return time directly corresponds to the time the part is available.
5. Holding costs increase with the increasing material value of the part, work-in-process, finished product at the manufacturer, and the product at the customer's location.
6. Vehicle idle costs are ignored.
7. There are no machine breakdowns and no defective parts or products.
8. Setup durations are assumed to be fixed and independent of both production order and batch volume.
9. Part pickup and product delivery are carried out using separate vehicles, where Delivery to the customer uses a single vehicle, and the capacity of the pickup vehicle exceeds the total product demand.

The objective function in this study is to minimize the total relevant costs, comprising holding costs, pickup costs, delivery costs, and rack procurement costs. Holding costs are derived from the AFT; pickup costs are based on the total distance traveled by all vehicles collecting parts; delivery costs are influenced by delivery frequency; and rack procurement costs depend on the number of production batches formed. Therefore, the objective function is formulated by multiplying the four remaining terms by their respective costs as shown in Table 2, and adding the pickup, delivery, and rack procurement costs, resulting in the following objective function:

$$\begin{aligned}
& \text{Min TC} \\
& = \left(\sum_{f \in \Omega} \sum_{g \in \Omega, g \neq f} \sum_{a \in A} c^k J r_{(f,g)} X_{(f,g,a)} \right) \\
& + \left(\sum_{i \in I} \sum_{f \in \Phi} c^s Q_{(i)} \left(B p_{(i)} - \sum_{a \in A} \delta_{(i,f,a)} \right) \right) \\
& + \left(\sum_{i \in I} c^w t Q_{(i)}^2 \right) \\
& + \left(\sum_{i \in I} c^r Q_{(i)} (B d_{2(i)} - C p_{(i)}) \right) \\
& + \left(\sum_{h \in H} c^t J p_{(h)} \right) \\
& + \left(\sum_{i \in I} \sum_{r \in R} c^m q_{(i,r)} (d_{(r)} - C d_{(i)}) \right) \\
& + \left(\sum_{i \in I} c^c O_{(i)} \right)
\end{aligned} \tag{7}$$

Subject to:

Demand allocation constraints:

$$\sum_{i \in I} q_{(i,r)} = Q_r, \quad \forall r \in R$$

$$\sum_{i \in I} Q_{(i)} = n$$

$$\sum_{r \in R} q_{(i,r)} = Q_{(i)}, \quad \forall i \in I$$

$$q_{(i,r)} \leq M v_{(i,r)}, \quad \forall i \in I, \forall r \in R$$

Pickup routing constraints:

$$\sum_{f \in \Omega, f \neq e} X_{(f,e,a)} = \sum_{g \in \Omega, g \neq e} X_{(e,g,a)}, \quad \forall e \in \Phi, \forall a \in A$$

$$\sum_{g \in \Omega, g \neq f} X_{(f,g,a)} = \theta_{(f,a)}, \quad \forall f \in \Phi, \forall a \in A$$

$$X_{(f,f,a)} = 0, \quad \forall f \in \Phi, \forall a \in A$$

$$\sum_{f \in \Omega, f \neq g} \sum_{a \in A} X_{(f,g,a)} = \sum_{a \in A} \theta_{(g,a)}, \quad \forall g \in \Phi$$

$$\sum_{g \in \Phi} X_{(N_0,g,a)} = Y_{(a)}, \quad \forall a \in A$$

$$\sum_{f \in \Phi} X_{(f,N_0,a)} = Y_{(a)}, \quad \forall a \in A$$

$$Z_{(N_0,a)} = 0, \quad \forall a \in A$$

$$\sum_{a \in A} U_{(i,f,a)} = O_{(i)}, \quad \forall i \in I, \forall f \in \Phi$$

$$U_{(i,f,a)} \leq \theta_{(f,a)}, \quad \forall i \in I, \forall f \in \Phi, \forall a \in A$$

$$Z_{(f,a)} = \sum_{i \in I} U_{(i,f,a)} Q_{(i)}, \quad \forall f \in \Phi, \forall a \in A$$

$$\sum_{f \in \Omega} Z_{(f,a)} \leq W Y_{(a)}, \quad a \in A$$

Production scheduling constraints:

$$B p_{(i)} = B s_{(i)} + s O_{(i)}, \quad \forall i \in I$$

$$C p_{(i)} = B p_{(i)} + Q_{(i)} t, \quad \forall i \in I$$

$$Q_{(i)} \leq c O_{(i)}, \quad \forall i \in I \tag{25}$$

$$O_{(i)} \leq Q_{(i)}, \quad \forall i \in I \tag{26}$$

$$O_{(i)} \geq O_{(i+1)}, \quad \forall i \in I, i < |I| \tag{27}$$

$$C p_{(i)} \leq B s_{(i-1)}, \quad \forall i \in I, i > 1 \tag{28}$$

Delivery batching constraints:

$$\sum_{h \in H} X p_{(h,i)} = O_{(i)}, \quad \forall i \in I \tag{29}$$

$$J p_{(h)} \leq \sum_{i \in I} X p_{(h,i)}, \quad \forall h \in H \tag{30}$$

$$\sum_{i \in I} X p_{(h,i)} \leq k J p_{(h)}, \quad \forall h \in H \tag{31}$$

$$B d_{(h)} \leq M J p_{(h)}, \quad \forall h \in H \tag{32}$$

$$S p_{(h)} \leq B d_{(h-1)}, \quad \forall h \in H, h > 1 \tag{33}$$

Timing constraints:

$$L_{(g,a)} \leq M \theta_{(g,a)}, \quad \forall g \in \Phi, \forall a \in A \tag{34}$$

$$L_{(g,a)} \geq \left(\frac{J R_{(N_0,g)}}{V l} + p \right) X_{(N_0,g,a)}, \quad \forall g \in \Phi, \forall a \in A \tag{35}$$

$$L_{(g,a)} \geq L_{(f,a)} + p + \left(\frac{J R_{(f,g)}}{V l} \right) - M(1 - X_{(f,g,a)}), \quad \forall f, g \in \Phi, f \neq g, \forall a \in A \tag{36}$$

$$W k_{(a)} = \sum_{f \in \Phi} \left(L_{(f,a)} X_{(f,N_0,a)} + \frac{J R_{(f,N_0)}}{V l} X_{(f,N_0,a)} \right), \quad \forall a \in A \tag{37}$$

$$L_{(f,a)} \leq W k_{(a)}, \quad \forall a \in A \tag{38}$$

$$\delta_{(i,f,a)} = W k_{(a)} U_{(i,f,a)}, \quad \forall i \in I, \forall f \in \Phi, \forall a \in A \tag{39}$$

$$\delta_{(i,f,a)} \leq B p_{(i)}, \quad \forall i \in I, \forall f \in \Phi, \forall a \in A \tag{40}$$

$$S p_{(h)} = B d_{(h)} + 2 v J p_{(h)}, \quad \forall h \in H \tag{41}$$

$$B d_{2(i)} = \sum_{h \in H} B d_{(h)} X p_{(h,i)}, \quad \forall i \in I \tag{42}$$

$$B d_{2(i)} \geq C p_{(i)} - M(1 - O_{(i)}), \quad \forall i \in I \tag{43}$$

$$C d_{(i)} = \sum_{h \in H} (S p_{(h)} X p_{(h,i)}) - v O_{(i)}, \quad \forall i \in I \tag{44}$$

$$C d_{(i)} \leq d_r + M(1 - v_{(i,r)}), \quad \forall i \in I, \forall r \in R \tag{45}$$

Equation (7) represents the objective function for minimizing total relevant costs, which include pickup costs, holding costs (for parts, WIP, finished goods, and products arriving before the due date), delivery costs, and rack procurement costs. Equation (8) ensures that all customer demands are met by guaranteeing that the total quantity of products allocated from all production batches (i) for due date r exactly matches the quantity requested by the customer for that order $Q_{(r)}$. Equation (9) ensures that the quantity of products manufactured equals the total customer demand. Equation (10) ensures that the total quantity of products from a batch allocated to meet requirements across all due dates r is equal to the size of that production batch. Equation (11) ensures that if a product from batch i is shipped for order r ($q_{ir} > 0$, then $v_{(i,r)} = 1$, and vice versa. Equation (12) establishes route continuity, meaning that if vehicle a visits supplier e , the same vehicle must also depart from supplier e to proceed to the next supplier. Equation (13) links the routing variable $X_{(f,g,a)}$ with the binary visit variable $\theta_{(f,a)}$; if vehicle a leaves node f for another node, then $\theta_{(f,a)} = 1$. Equation (14) ensuring no self-loops: the vehicle must not go from f to f . Equation (15)

governs global visit consistency, integrating physical vehicle routing decisions with pickup status. Equations (16) and (17) guarantee that if vehicle a is used ($Y_{(a)} = 1$), it must depart from the manufacturer (N_0) to a supplier and return to the manufacturer. Equation (18) specifies that no part pickup occurs at the manufacturer. Equation (19) ensures that, for each batch, the pickup of parts from each supplier is assigned to only one vehicle. Equation (20) restricts the pickup of part for batch i from supplier f by vehicle a to cases where vehicle a actually visits supplier f . Equation (21) calculates the total number of parts transported by vehicle a when collecting parts from supplier f by summing the sizes of all production batches scheduled to be picked up by that vehicle from that supplier. Equation (22) ensures that the quantity of parts transported from all suppliers ($z_{(f,a)}$) by the pickup vehicle a does not exceed its capacity. Equation (23) states that the production start time for batch i ($Bp_{(i)}$) is the machine setup start time ($Bs_{(i)}$) plus the setup duration (s), provided the batch is actively being produced $O_{(i)} = 1$. Equation (24) states that the production completion time for batch i ($Cp_{(i)}$) is calculated as the production start time ($Bp_{(i)}$) plus the total processing time, specifically, the batch size $Q_{(i)}$ multiplied by the processing time per unit (t). Equation (25) ensures that the active batch size does not exceed the capacity. Equation (26) ensures that the minimum active batch size is 1. Equation (27) is a symmetry-breaking delimiter that requires active batches to occupy consecutive indices starting from the first index. Equation (28) enforces backward scheduling: a batch with a higher index, i , must complete production $Cp_{(i)}$ before the batch with the preceding index, $(i - 1)$, begins its setup process $Bs_{(i)}$, ensuring that no two batches are produced simultaneously on the same machine. Equation (29) ensures that each produced batch i ($O_{(i)} = 1$) is loaded into exactly one delivery h , and that if a batch is not produced, it is not shipped. Equation (30) ensures that delivery h is active ($Jp_{(h)} = 1$) if at least one production batch i is loaded onto that delivery ($Xp_{(h,i)} = 1$). Equation (31) specifies the delivery vehicle's capacity limit. Equation (32) represents the departure time ($Bd_{(h)} > 0$), which is active if the delivery itself is active. Equation (33) represents backward scheduling, requiring that a delivery with a higher index number (h) must return to the manufacturer ($Sp_{(h)}$) before a delivery with a lower index number departs. Equation (34) ensures that $L_{(g,a)}$ takes a positive value if vehicle a visits node g ; otherwise, $L_{(g,a)}$ equals 0. Equation (35) ensures the vehicle departs from the manufacturer. Equation (36) serves to eliminate sub-tours and accounts for cumulative time; if vehicle a moves from supplier f to supplier g , the departure time from supplier g ($l_{(g,a)}$) must be greater than or equal to the departure time from supplier f plus the part loading time p and the travel time from f to g . Equation (37) indicates the time of the pickup vehicle's return to the manufacturer. Equation (38) ensures that the departure time from the supplier does not exceed the pickup vehicle's return time to the manufacturer. Equation (39) synchronizes part arrival times, linking pickup and production. Equation (40) limits the part arrival time for a batch to occur before that batch's production start time. Equation (41) specifies that the time at which the delivery truck h returns to

the manufacturer is the truck's departure time ($Bd_{(h)}$) plus the round-trip travel time to the customer ($2 \times \text{travel time}, v$). Equation (42) is an auxiliary variable used to record the departure time of the truck transporting batch i . Equation (43) guarantees that the delivery time of batch i occurs after the batch has finished processing. Equation (44) represents an auxiliary variable used to record the arrival time of batch i at the customer. Equation (45) ensures that if batch i is allocated to fulfill customer order r ($v_{(i,r)} = 1$), then the product's arrival time at the customer $Cd_{(i)}$ must not exceed the order's due date.

3. Solution Method

The formulated model is solved using several approaches:

3.1. Analytical approach

Analytical model solving aims to find an optimal solution as an explicit function that satisfies all equations in the model. For optimization models, analytical solutions are obtained using differentiation techniques. According to [30], for a model to be solved analytically using derivatives, it must satisfy several mathematical conditions:

- The model function must be continuous
The model function must not exhibit jumps or extreme changes within its domain interval. Theoretically, if a function is differentiable at a specific point, it is necessarily continuous at that point.
- The model function should be differentiable
The model must have a derivative over its domain or interval, which implies the difference ratio must have a limit, and that limit must be a real number.
- The model has a specific focus for the critical point
Extreme values (maximum or minimum) for an optimization model can only occur at the function's critical points. This critical point theorem states that these critical points are:
 - The endpoints of the interval of the function's domain.
 - Local extreme values: points x where $f'(x) = 0$.
 - Singular points where the derivative of the function is not defined.
- The model has an equation of the differential type that can be integrated exactly
If the model is a differential equation (including derivatives), it can be solved analytically only if it is in a standard form that allows exact integration.

This criterion makes the model in this study analytically unsolvable, because:

- It uses binary variables.
For a function to be solved by differentiation, it must be continuous and differentiable on its domain. The limit $\lim_{h \rightarrow 0} \left(\frac{f(x+h) - f(h)}{h} \right)$ requires x to have continuous values as h approaches 0. In this model, many variables are binary, so their value domain is not an interval. Note that there is no notion of continuous

values in between 0 and 1, so the derivative limit at the points in between does not exist.

- b. No critical points can be derived.

The Critical Point Theorem states that the critical points (maximum or minimum) of a function can only occur at interval endpoints, stationary points, or singular points. The objective function cannot be differentiated with respect to binary variables because the model is combinatorial (discrete) in nature; consequently, no stationary point where the derivative equals zero can be found.

- c. The model falls into the category of mixed-integer nonlinear programming (MINLP), and MINLP problems are classified as NP-hard, making them difficult to solve analytically.

Since analytical methods cannot resolve the problem presented by this model, it will be solved exactly using a solver.

3.2. Exact approach using the Gurobi solver

The mathematical model was solved exactly using Python 3.14.4 via the Gurobi Python interface within the Visual Studio Code environment. Computations were performed using Gurobi Optimizer 13.0.2 on a 64-bit Windows 10 PC equipped with an Intel Core i7-12700K processor (12 cores, 20 threads) and 8 GB of DDR4 RAM. During the solution process, the model iterates through the datasets and parameters. The set of pickup vehicles, all location points, supplier locations, and customer due dates are all known. The set of maximum deliveries is derived from the model's inter-constraint relationships linking the maximum due date to the delivery start time, resulting in a maximum number of deliveries equal to $\left\lceil \left(\frac{d_{(r)max} - v}{2v} \right) + 1 \right\rceil$. Since the maximum capacity per shipment is k batches, the maximum quantity becomes $\left(k * \left\lceil \left(\frac{d_{(r)max} - v}{2v} \right) + 1 \right\rceil \right)$. When considering batch capacity, the minimum number of production batches is $\left(\frac{n}{c} \right)$ and the maximum is n (where each production batch has a size of 1); thus, the maximum number of production batches is derived based on the relationship with shipments and batch capacity, resulting in a maximum number of production batches of $min = \left\{ n, k * \left\lceil \left(\frac{d_{(r)max} - v}{2v} \right) + 1 \right\rceil \right\}$. When the Gurobi solver runs with production and shipment batch ranges from the minimum to the maximum values, it gets stuck and fails to find an optimal solution. This phenomenon of stalling is due to several reasons:

- a. Computational Complexity

The mathematical model in this study is a non-convex MINLP (Mixed-Integer Nonlinear Programming) problem, making it unsuitable for Gurobi, which is primarily designed for MILP and convex optimization. The model's nonlinearity arises from multiplication operations between decision variables, manifesting as quadratic functions (between discrete variables) and bilinear functions (between continuous and discrete variables). Furthermore, the model is non-convex due to both the geometry of the objective function caused by the bilinear multiplication of time and quantity variables and the presence of nonlinear constraints that render the solution search space (feasible region) non-convex.

- b. Combinatorial explosion on discrete variables:

The exponential increase in discrete variables places a heavy load on the solver's memory because the Branch and Bound algorithm must evaluate many combinations. Consequently, the solver encounters a computational bottleneck because it must recompute matrix algebraic operations at each node, drastically slowing iteration speed.

The following example shows how to solve a problem using the Gurobi solver. Some parameter data use the same values as in [28], whereas sets H and I use the values derived in this subsection (3.2).

Table 3. Parameter Data

$W = 300$	$k = 3$	$c^c = 25$
$M = 200$	$t = 0.5$	$c^t = 50$
$Vl = 10$	$s = 2$	$c^r = 20$
$c = 20$	$v = 20$	$c^k = 10$
$c^m = 30$	$c^s = 1$	$c^w = 15$
$p = 1$		

The data on distances between suppliers can be seen in Table 4.

Table 4. Distance Matrix (km)

Node	N_0	N_1	N_2	N_3	N_4	N_5	N_6
N_0	9999	136	54	139	101	62	144
N_1	136	9999	73	64	107	86	102
N_2	54	73	9999	93	128	50	129
N_3	139	64	93	9999	63	105	141
N_4	101	107	128	63	9999	149	116
N_5	62	86	50	105	149	9999	96
N_6	144	102	129	141	116	96	9999

Table 5 shows the calculation results for several datasets run using Gurobi and interrupted at 10.800 seconds.

Table 5. Experimental Result

Data set	$Q_{(r)}$	$d_{(r)}$	Best Feasible TC	Gap (%)
1	$Q_{(1)}: 50, Q_{(2)}: 70$	$d_{(1)}: 150, d_{(2)}: 200$	82.610	157
2	$Q_{(1)}: 50, Q_{(2)}: 80$	$d_{(1)}: 175, d_{(2)}: 200$	129.625	129
3	$Q_{(1)}: 90, Q_{(2)}: 60$	$d_{(1)}: 180, d_{(2)}: 200$	169.665	92,4
4	$Q_{(1)}: 45, Q_{(2)}: 55, Q_{(3)}: 75$	$d_{(1)}: 150, d_{(2)}: 170, d_{(3)}: 200$	184.762,5	88,8
5	$Q_{(1)}: 80, Q_{(2)}: 50, Q_{(3)}: 70$	$d_{(1)}: 150, d_{(2)}: 170, d_{(3)}: 200$	232.550	65,1

Table 5 shows that, after 10.800 seconds of computation, an optimal solution has not yet been obtained for all datasets; the gap between the best feasible solution and the lower bound remains significant. Therefore, a metaheuristic algorithm is required to solve the integration problem addressed in this study.

3.3. Genetic Algorithm

This study proposes a genetic algorithm (GA) to solve the integrated model for pickup, production, and batch delivery. The GA enables the simultaneous resolution of problems involving diverse types of decision variables through an evolutionary process, eliminating the need to decompose the problem into sub-problems that must be solved sequentially. The decisions generated by this model are both quantitative and combinatorial. In production, the combinatorial decision involves determining the number of active batches out of the

maximum possible, while the allocation decision for each batch is quantitative. For delivery, the decision to allocate batches to shipments is combinatorial, whereas the decision on shipment delays for each batch is quantitative. Unlike production and delivery, decisions about grouping supplier pickup tasks to vehicles and sequencing supplier visits are combinatorial. Because of these differing characteristics, the gene representing the supplier pickup sequence is encoded as a permutation, while the other genes are encoded as real numbers. Figure 3 shows the flowchart of the genetic algorithm used.

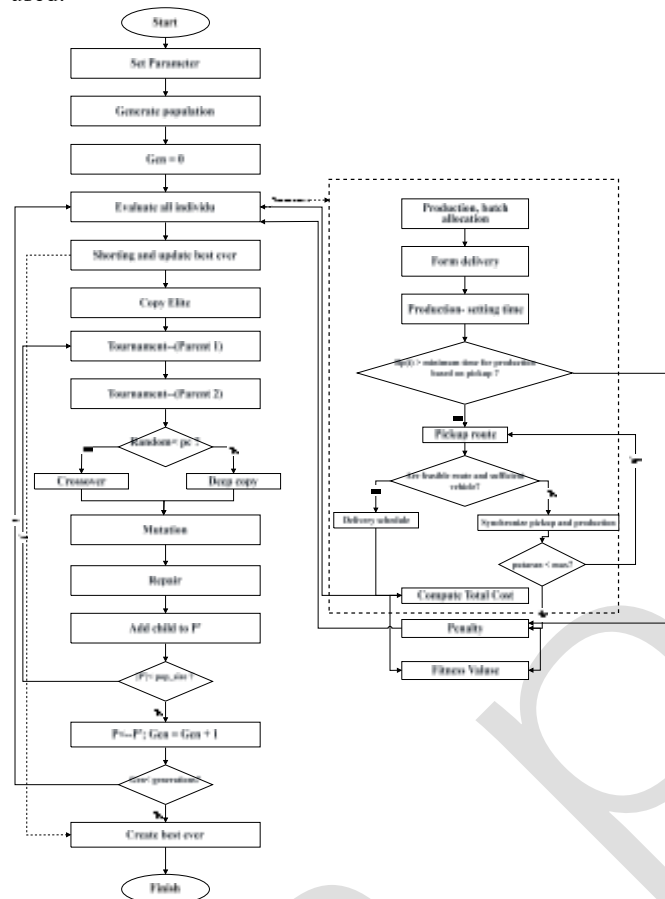


Figure 3. Flowchart of GA

3.3.1. Chromosome representation

Each solution to the integration problem in this study is represented by a chromosome containing instructions for constructing the solution, while the solution value is determined through decoding. Table 6 shows the chromosome representation used in the GA for this study.

Table 6. Chromosome Representation

Gen	Representation	Decision	Solution Representation
A	Matrix of real numbers	Combinatorics and quantities	Number and size of each active batch, batch allocation for each due date
D_cut	Real-valued vector	Combinatorics	Assignment from production batch to shipment
D_tau	Real-valued vector	Quantities	Delay during production batch delivery
E_tour	Permutation	Combinatorics	Assignment of production batches to suppliers

E_Cut	Real-valued vector	Combinatorics	Assignment of tour results to the pickup vehicle
-------	--------------------	---------------	--

3.3.2. Decoding

Decoding maps the chromosome to a solution representation by making decisions about integrating pickup, production, and batch delivery, and calculates the objective function value, which minimizes total relevant costs.

a. Section A: Production Batch Filling

Production batch filling is carried out by first calculating the batch size sequentially, starting with the first batch, based on the gene value multiplied by the production batch capacity. Since each production batch can satisfy demands associated with multiple due dates, the allocation for each due date is calculated based on the proportion of the gene value assigned to that due date multiplied by the size of the active production batch. This process continues until all demands across all due dates are met, with the total number of filled batches representing the number of active batches.

b. Section C: Production scheduling using a backward approach

Section C translates backward scheduling based on production completion deadlines derived from the production batch assignments generated in Section A and the delivery requirements (Output of Section D) into specific start times for setup, production start, and production completion for each production batch.

c. Section E: Planning the pickup route

This section establishes pickup routes and vehicle assignments for parts required to produce Section A. It begins by discarding inactive batches, thereby creating pairs of active batches and their corresponding suppliers. Each pair is assigned an E_cut value generated from a random number; the load and E_cut value of each batch-pair sequence are then evaluated. If either criterion is met, the sequence is split into a group, with each group forming a single route. These generated routes are subsequently refined using a local search. Following the local search, trip groups are formed; if the number of groups is less than the available pickup vehicles, a "split milk run" procedure is applied, whereas if the number exceeds the available vehicles, a merge milk run repair is performed. The results of the merge milk run repair are then subjected to the split milk run repair process. Finally, after the split milk run repair, the routes are further refined to explore reassigning suppliers to different trips, resulting in the final set of trips.

d. Section D:

This section organizes customer delivery groups and determines the departure time for each shipment. It utilizes two chromosomes: D_cut and D_tau. D_cut governs trip formation based on the production batch sequence; trips form when delivery capacity is reached, the D_cut value at a given position exceeds the trip-splitting threshold, or the batch is the final one. Based on the formed delivery trips, the latest

completion time (determining the earliest possible departure) and the earliest due date among the batches in a trip (setting the upper limit for departure, adjusted for vehicle round-trip time) are identified; D_{τ} then determines the actual departure time within these lower and upper limits.

3.3.3. Synchronization between the stages

- Synchronization of production and Delivery**
Because a backward scheduling approach is used, the delivery departure time is determined from the relationship between the due date and the delivery duration for a single vehicle. This decision serves as the basis for establishing the production schedule in Section C.
- Synchronization of pickup and production**
The mechanism for synchronizing pickup and production involves an iterative process because pickup operations depend on the batch quantity (B_p) derived from production scheduling, while the actual required B_p is only known once the pickup route is established. This differs from delivery synchronization, where requirements can be calculated from input data and managed proactively via established limits. If Section E fails, the procedure determines the actual lower time limit based on the established route, delaying the production start time. As the production start time cannot be altered directly and depends solely on the delivery trip configuration, adjustments are made by reorganizing delivery trips through merging, splitting, or shifting boundaries.

3.4. Fitness function

The fitness value of a generated chromosome is calculated from the total cost derived during decoding, comprising pickup costs; holding costs for parts, work-in-process, and finished goods (at both the manufacturer and customer sites); shipping costs; and rack procurement costs. Chromosomes that violate constraints incur penalties with both fixed and variable components, with the latter determined by the number of violations. These penalties adjust the total cost value based on feasibility checks performed at various stages of the decoding procedure, covering factors such as demand fulfillment, production chain feasibility, minimum pickup time limits, vehicle capacity, part arrival time windows, delivery deadlines, and limits on the number of delivery trips.

3.5. Genetic Algorithm Operators

3.5.1. Generation of the initial population

The initial population is generated randomly. The row matrices in Section A are divided into two groups based on the estimated minimum number of batches required to meet the due date. The first group is generated with a dominant column, whereas the second group is generated with balanced weights across columns. Cut genes are generated from Beta distributions with parameters skewed in opposite directions: shipment trip cut genes tend toward low values, resulting in infrequent cuts and trips that are typically fully loaded, while pickup group cut genes tend toward high values, resulting in frequent cuts.

3.5.2. Selection, crossover, and mutation

Parent selection is carried out via a tournament mechanism conducted separately to obtain a pair of parents. Crossover for real-valued genes is performed using simulated binary crossover with probability p_c , whereas for permutation genes, the order crossover mechanism is employed. This approach is chosen because order crossover guarantees that each (batch, supplier) pair appears exactly once. In contrast, simulated binary crossover produces intermediate values that could result in duplicates or invalid indices if applied to permutations. Mutation is applied to each offspring: real-valued genes are mutated by adding normally distributed noise (with a mean of zero and standard deviation σ) at a probability of p_m per gene, followed by clipping to the defined range limits, while permutation genes are mutated by swapping positions at a probability of p_{mp} per gene.

3.5.3. Elitism

Elitism is implemented by copying the individuals with the lowest total cost values. These individuals are copied intact into the next generation's population without undergoing any refinement procedures. Meanwhile, the best solution found throughout the process is recorded outside the population and returned as the final output, ensuring the best solution is preserved across generations.

3.5.4. Repair Procedure

Repairs are made at both the gene level and the decoding output level. At the gene level, adjustments eliminate zero-value proportion entries for due dates, ensure total production capacity meets all demand, and consolidate partially filled batches into full batches. At the decoding output level, four procedures are executed: merging pickup trips, splitting pickup trips, and reassigning suppliers to different pickup trips, because they require information available only at that stage, such as distances between suppliers, batch production start times, and vehicle capacities. These four procedures do not alter the chromosome itself; consequently, the resulting changes are not inherited.

3.6. Numerical Experience

The following compares the calculation results obtained using the exact method and the genetic algorithm. We performed the exact-method calculation using the Gurobi solver and ran it for 10.800 seconds, using the same dataset as in Table 5. Table 7 compares the results.

Table 7. Comparison of calculation TC using the exact method and GA

Dataset Number	Exact	GA	Computation time GA (seconds)	Gap (%)
1	82.610	82.610	60,5	0
2	129.625	130.260	57	0,4
3	169.665	169.200	74,4	-0,2
4	184.762,5	190.952,5	91,2	3,3
5	232.550	233.670	121	0,4

Table 7 shows that the cost gap between the GA solution and the best feasible solution obtained by the solver is relatively small. The GA can generate a solution close to the best feasible solution found by the exact method while requiring significantly less computation time than the solver. The following is a detailed example of the solution for dataset 5,

solved using exact methods and GA. Table 8 shows the exact solutions for production decisions.

Batch (i)	$Q_{(i)}$	$BS_{(i)}$	$Bp_{(i)}$	$Cp_{(i)}$	$q_{(i,r)}$
1	20	168	170	180	(1,3):20
2	20	156	158	168	(2,3):20
3	20	144	146	156	(3,3):20
4	20	128	130	140	(4,2):20
5	20	116	118	128	(5,2):10,(5,3):10
6	20	104	106	116	(6,2):20
7	20	88	90	100	(7,1):20
8	20	76	78	88	(8,1):20
9	20	64	66	76	(9,1):20
10	20	48	50	60	(10,1):20

Solutions obtained using GA for production decisions are presented in Table 9.

Batch (i)	$Q_{(i)}$	$BS_{(i)}$	$Bp_{(i)}$	$Cp_{(i)}$	$q_{(i,r)}$
1	20	168	170	180	(1,3):20
2	20	156	158	168	(2,3):20
3	20	144	146	156	(3,3):20
4	20	128	130	140	(4,2):10,(4,3):10
5	20	116	118	128	(5,2):20
6	20	104	106	116	(6,2):20
8	20	88	90	100	(8,1):20
9	20	76	78	88	(9,1):20
10	20	64	66	76	(10,1):20
11	20	48	50	60	(11,1):20

Solutions obtained using exact methods for delivery decisions are presented in Table 10.

Delivery	Production Batch	$Bd_{2(i)}$	$Cd_{(i)}$
1	$Q_{(3)}, Q_{(2)}, Q_{(1)}$	180	200
2	$Q_{(6)}, Q_{(5)}, Q_{(4)}$	140	160
3	$Q_{(9)}, Q_{(8)}, Q_{(7)}$	100	120
4	$Q_{(10)}$	60	80

Table 11 shows the GA solutions for delivery decisions.

Delivery	Production Batch	$Bd_{2(i)}$	$Cd_{(i)}$
1	$Q_{(11)}$	60	80
2	$Q_{(10)}, Q_{(9)}, Q_{(8)}$	100	120
3	$Q_{(6)}, Q_{(5)}, Q_{(4)}$	140	160
4	$Q_{(3)}, Q_{(2)}, Q_{(1)}$	180	200

Table 12 shows solutions obtained using exact methods for pickup decisions. Table 13 presents the GA solutions for pickup decisions. Figure 4 shows the integration Gantt chart generated using the exact method. Figure 5 shows the integration Gantt chart generated using the exact method.

Vehicle	Route	$\delta_{(i,f,a)}$	Supplier	Production batches handled	Quantity per batch	Load
1	$N_0 - N_4 - N_3 - N_1 - N_2 - N_5 - N_0$	146	N_1, N_2, N_3, N_4, N_5	$Q_{(3)}, Q_{(2)}, Q_{(1)}$	20	300
2	$N_0 - N_2 - N_1 - N_3 - N_4 - N_6 - N_5 - N_0$	90	$N_1, N_2, N_3, N_4, N_5, N_6$	$Q_{(7)}, Q_{(6)}$	20	240
3	$N_0 - N_5 - N_6 - N_0$	50	N_5, N_6	$Q_{(10)}, Q_{(9)}, Q_{(8)}$	20	120
4	$N_0 - N_5 - N_6 - N_4 - N_3 - N_1 - N_2 - N_0$	118	N_1, N_2, N_3, N_4, N_5	$Q_{(5)}, Q_{(4)}$	20	300
			N_6	$Q_{(5)}, Q_{(4)}, Q_{(3)}, Q_{(2)}, Q_{(1)}$	20	
5	$N_0 - N_4 - N_3 - N_1 - N_2 - N_0$	50	N_1, N_2, N_3, N_4	$Q_{(10)}, Q_{(9)}, Q_{(8)}$	20	240

Vehicle	Route	$\delta_{(i,f,a)}$	Supplier	Production batches handled	Quantity per batch	Load
1	$N_0 - N_5 - N_6 - N_4 - N_3 - N_1 - N_2 - N_0$	146	N_2, N_5, N_6	$Q_{(3)}, Q_{(2)}, Q_{(1)}$	20	300
			N_3, N_4	$Q_{(3)}, Q_{(2)}$	20	
			N_1	$Q_{(2)}, Q_{(1)}$	20	
2	$N_0 - N_5 - N_2 - N_1 - N_3 - N_4 - N_0$	118	N_3, N_4	$Q_{(5)}, Q_{(4)}, Q_{(1)}$	20	260
			N_2, N_5	$Q_{(5)}, Q_{(4)}$	20	
			N_1	$Q_{(5)}, Q_{(4)}, Q_{(3)}$	20	
3	$N_0 - N_6 - N_5 - N_0$	50	N_5, N_6	$Q_{(11)}, Q_{(10)}, Q_{(9)}$	20	120
4	$N_0 - N_2 - N_1 - N_3 - N_4 - N_6 - N_5 - N_0$	90	N_1, N_2, N_3, N_4, N_5	$Q_{(8)}, Q_{(6)}$	20	280
			N_6	$Q_{(8)}, Q_{(6)}, Q_{(5)}, Q_{(4)}$	20	
5	$N_0 - N_2 - N_1 - N_3 - N_4 - N_0$	50	N_1, N_2, N_3, N_4	$Q_{(11)}, Q_{(10)}, Q_{(9)}$	20	240

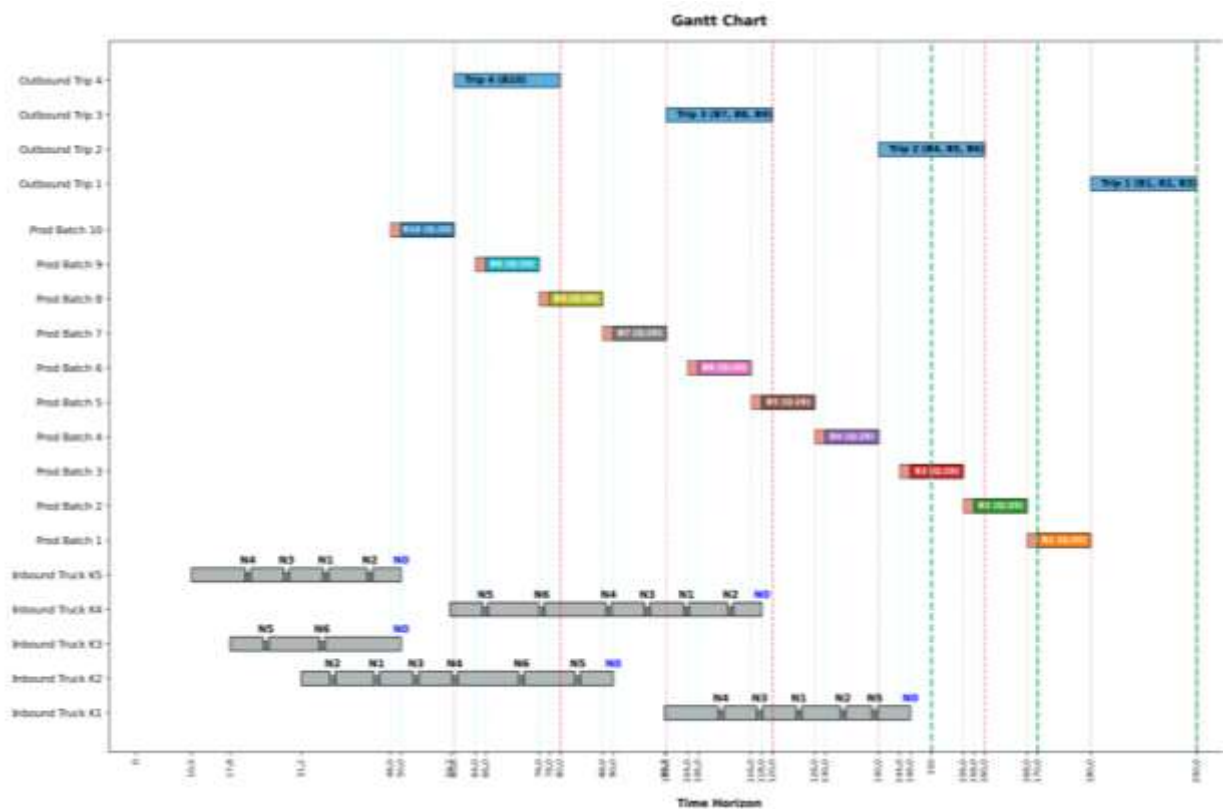


Figure 4. Gantt chart using an exact method

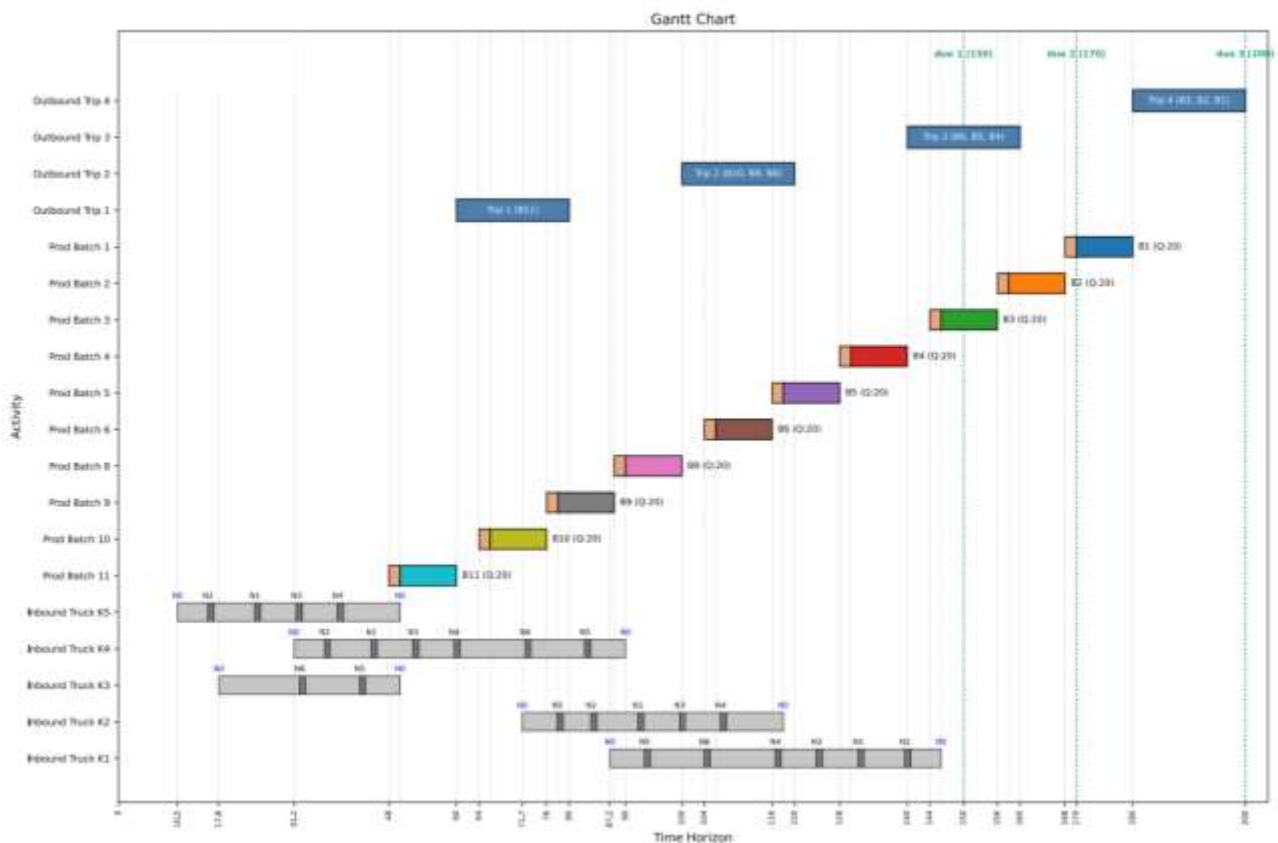


Figure 5. Gantt chart using GA

For dataset 5, the solution obtained differs by 0,4% in cost between the exact method and the GA. This cost difference

stems from raw material holding costs, resulting from differences in the batches served on each pickup route.

3.7. Tuning Parameters

Parameter tuning was performed using the Taguchi method. The GA parameters employed were crossover probability ($pc = 0,6$), real-gene mutation probability ($pm = 0,08$), permutation mutation probability ($pmp = 0,008$), mutation deviation ($\sigma = 0,4$), tournament size (5), and elite count (5); each parameter was tested at three levels, potentially yielding 729 parameter combinations. However, as this number was too large for practical testing, an orthogonal array was utilized for the tuning process; specifically, an L27 array was used for the six parameters, each with three levels. Complete parameter data is presented in Table 14:

Table 14. GA Parameters configuration

Parameter	Level 1	Level 2	Level 3	Initial configuration
<i>pc</i>	0,4	0,6	0,8	0,6
<i>pm</i>	0,04	0,08	0,12	0,08
<i>pmp</i>	0,004	0,008	0,016	0,008
<i>sigma</i>	0,2	0,4	0,6	0,4
<i>tournament</i>	3	5	7	5
<i>elitism</i>	3	5	7	5

The GA parameter tuning process was carried out on dataset 4 using 5 seeds, resulting in 135 tuning data points. Based on this tuning data, the S/N ratio was then calculated:

$$S/N = -10 \log_{10} \left(\left(\frac{1}{n} \right) \times \sum (TC_i^2) \right) \quad (46)$$

Where n is the number of replications (seeds). The parameter tuning results for dataset 4 are shown in Table 15.

Table 15. Result of Parameter Tuning

Parameter	Average S/N Level 1	Average S/N Level 2	Average S/N Level 3	Range	Best Parameters value
<i>pc</i>	-105,6379	-105,6358	-105,6323	0,0056	0,8
<i>pm</i>	-105,6316	-105,6348	-105,6397	0,0081	0,04
<i>pmp</i>	-105,6394	-105,6330	-105,6337	0,0064	0,008
<i>sigma</i>	-105,6357	-105,6367	-105,6335	0,0032	0,6
<i>tournament</i>	-105,6384	-105,6373	-105,6303	0,0080	7
<i>Elitism</i>	-105,6344	-105,6383	-105,6333	0,0051	7

From Table 15, it can be observed that, based on range values, the largest and smallest ranges are produced by the real-coded gene mutation probability and mutation standard deviation parameters, with a relatively small difference between them. On the other hand, regarding the S/N values for the L27 array, the S/N range is 0,0413, with a standard deviation of 0,0109. This range is relatively small, indicating that the tested genetic algorithm is not sensitive to changes in the parameter values under examination. Table 13 also shows the optimal parameter values; notably, when these six values are combined, the resulting combination is not among the L27 array combinations.

3.8. Validation

The GA parameter combinations resulting from the tuning process were tested using five datasets, with each dataset replicated 30 times. The combination of GA parameters resulting from the tuning process was tested using five datasets, with each dataset replicated 30 times. The validation results are presented in Table 16.

Table 16. Validation

Dataset	Na	Nt	Ge (%)	Gt (%)
1	30	30	0,003	0,048
2	30	30	0,138	0,150
3	30	30	-0,252	-0,262
5	26	21	2,646	5,611
4	30	30	3,569	3,433

*Na: Number of feasible solutions for the initial parameter configuration

*Nt : Number of feasible solutions for the tuning parameter configuration

*Ge : Average total cost gap between the exact method and the initial GA

*Gt : Average total cost gap between the exact method and the tuning GA

Based on Table 16, it can be seen that the parameter configurations derived from tuning, the initial settings, and the Taguchi method all yielded infeasible solutions for data set 5; however, the Taguchi configuration resulted in a higher number of infeasible solutions. Furthermore, the average Total Cost (TC) gap between the exact method and the GA was smaller than the gap between the exact method and the GA with the Taguchi configuration; consequently, the initial parameter configuration was selected for the GA.

3.9. Statistical Testing

Statistical tests were conducted to evaluate the GA's performance based on the consistency of the results obtained given the stochastic nature of the GA and to compare the total costs generated by the solver and the GA. Testing was performed using the five datasets listed in Table 5, with the initial GA configuration replicated 30 times. The statistical tests employed included:

3.9.1. Normality Test (Shapiro-Wilk)

This test is conducted to determine whether the data are normally distributed, using a significance level of $\alpha = 0,05$. The hypotheses for the Shapiro-Wilk test are:

H_0 The total cost data in the dataset i are normally distributed.

H_1 The total cost data in the dataset i are not normally distributed.

Hasil uji statistik ditunjukkan pada Tabel 17:

Table 17. Result of Shapiro-Wilks test

Data set	Number of data	W-Statistic	P-value	Decision	Advanced statistical tests
1	30	0,4044	$5,98 \times 10^{-10}$	H_0 rejected	Wilcoxon 1-sampel
2	30	0,6127	$1,05 \times 10^{-7}$	H_0 rejected	Wilcoxon 1-sampel
3	30	0,5135	$7,46 \times 10^{-9}$	H_0 rejected	Wilcoxon 1-sampel
4	30	0,9672	0,4661	H_0 accepted	uji t 1-sampel
5	26	0,2731	$2,52 \times 10$	H_0 rejected	Wilcoxon 1-sampel

Testing of the five datasets revealed that datasets 1, 2, 3, and 5 are not normally distributed, whereas dataset 4 is normally distributed; consequently, different follow-up tests are required for these two groups of data.

3.9.2. Comparison test of GA and Solver solutions

This test is a follow-up to the Shapiro-Wilk test. The hypotheses used for data set 4 are:

H_0 The average total cost resulting from the GA is equal to the total cost from the solver.

H_1 The average total cost resulting from the GA is not equal to the total cost from the solver

Meanwhile, the hypothesis for the one-sample Wilcoxon test is:

H_0 The median total cost resulting from the GA is equal to the total cost from the solver

H_1 The median total cost resulting from the GA is not equal to the total cost from the solver

The results of the comparative test between the GA and Solver solutions are shown in Table 18:

Table 18. Comparative solution between GA and Solver

Data set	Number of data	Statistic	P-value	Decision
1	30	$W = 0,0$	$1,96 \times 10^{-7}$	H_0 rejected
2	30	$W = 0,0$	$6,88 \times 10^{-7}$	H_0 rejected
3	30	$W = 0,0$	$1,34 \times 10^{-6}$	H_0 rejected
4	30	$t = 78,2232$	$2,73 \times 10^{-35}$	H_0 rejected
5	26	$W = 0,0$	$6,06 \times 10^{-6}$	H_0 rejected

From Table 18, it can be seen that H_0 is rejected for all five datasets because the p-value is less than 0,05. For dataset 4, the rejection of H_0 indicates that the mean total GA cost across the 30 seeds differs significantly from the solver's total cost value for that dataset. For datasets 1, 2, 3, and 5, the rejection of H_0 indicates that the median total GA cost across the 30 seeds for each dataset differs significantly from the solver's value for the respective dataset.

3.9.3. Descriptive Statistics

Descriptive statistics were used to demonstrate the magnitude of variation in GA results across different seeds, as indicated by the coefficient of variation (CV) values. Descriptive statistics are presented in Table 19.

Table 19. Descriptive statistic

Data set	Feasibility Rate (%)	mean	Standard deviation	CV (%)	Average Gap (%)
1	100	82.612,7	6,9	0,008	0,003
2	100	129.804,3	271,76	0,209	0,138
3	100	169.237,2	163,60	0,097	-0,252
4	100	191.356	461,68	0,241	3,569
5	86,7	238.704,0	18.118,14	7,590	2,646

Statistical tests and descriptive statistical calculations reveal a high level of consistency in the GA, as evidenced by the CV values across datasets 1 through 4. These CV values demonstrate that the algorithm's stochastic nature does not render its results random or unpredictable. When compared to the solutions produced by the solver, the GA outperforms it on dataset 3. This demonstrates that the GA approach is capable of finding a solution superior to the solver's best feasible solution while maintaining rapid computation times. For the

other datasets, the gap between the GA solution and the solver's best feasible result ranges from 0,003% to 3,569%, yet the GA achieves this with significantly faster computation times.

4. Result and Discussion

The developed mathematical model represents the problem of integrating pickup, production, and batch delivery. The resulting model falls into the category of mixed-integer nonlinear programming. The formulated mathematical model was first solved using the Gurobi solver with a computation time of 10.800 seconds per dataset. Among the five tested datasets, an optimal solution was not found for any of them; consequently, the status of the obtained solutions was classified as the "best feasible solution," with the smallest gap of 65,1% observed in dataset 5. The developed Genetic Algorithm (GA) was capable of finding solutions approaching or even surpassing the solver's best feasible solution, while requiring significantly less computation time. Convergence analysis of the GA revealed that dataset 3 reached a stable point the fastest (at the 12th generation), whereas dataset 5 was the slowest (at the 71st generation out of 100 generated generations).

Given the stochastic nature of the GA, different seed values could potentially yield different solutions; therefore, statistical testing was conducted by running the GA thirty times for each dataset, resulting in 150 data points. The coefficient of variation (CV) for four of the five datasets was below 0,25%, with a minimum of 0,008% for dataset 1, indicating that result variation across seeds was negligible. The highest CV occurred in dataset 5 for a specific seed value, and the number of feasible solutions for this dataset was only 26 out of 30 runs.

A fundamental point is that the comparison against the solver should not be interpreted as a comparison against the true optimum, but rather against the best feasible solution found within the 10,800-second runtime. The solver's solution gaps (ranging from 65,1% to 157,0%) were far larger than those of the GA; the GA produced solutions that were only slightly more costly than the best solutions found by the solver within the 10,800-second limit. This explains the advantage observed in dataset 3: with a gap solver of 92,4%, the 0,252% advantage is not an anomaly, but rather a logical consequence of comparing two methods that have yet to reach optimal conditions. The characteristics of the datasets influence the quality of the resulting solutions. Dataset 3, featuring two requests and loose deadlines, presents a relatively smooth solution space, as evidenced by its fastest convergence. In contrast, Dataset 4 comprises three requests with tight, due dates, increasing the interdependency of decisions, and is the only dataset with normally distributed data; this indicates that the algorithm failed to lock onto a single dominant solution, unlike with the other datasets. Dataset 5, with the highest number of requests and deadlines similar to Dataset 4, yields a limited feasible solution space, thereby creating a risk of infeasibility.

In practical terms, the significantly shorter computation times combined with solution quality that is equal to or better than other methods across three of the five datasets make the genetic algorithm a viable alternative for situations requiring rapid responses.

5. Conclusion

5.1. Concluding Remarks

Based on the testing of several datasets using the solver and the proposed algorithm, it can be concluded that:

- a. Successful system modelling and integration
The mathematical model developed in this study successfully addresses the integration problem in its entirety. Specifically, the model encompasses raw material pickup routing from multiple suppliers, batch production scheduling at the manufacturing facility, and delivery scheduling for finished products to a single customer. The model's capabilities were validated by the success of both the exact method and the proposed GA in finding feasible solutions across all five experimental datasets, which featured varying demand quantities and due dates.
- b. Computational time of proposed GA
The proposed GA algorithm is capable of generating feasible solutions that approach the best feasible solutions obtained via exact methods using a solver, while requiring significantly less time than the solver itself
- c. GA computation time trade-off
Based on Table 7, as the problem scale increases (i.e., a greater number of requests and due dates), the GA computation time also increases. The gap between the Total Cost (TC) produced by the GA and the best feasible TC obtained by the solver likewise widens.

5.2. Limitation and Future Research

5.2.1. Research limitation

Although the developed model can address the problem of integrating pickup, production, and batch delivery, several limitations should be noted as a basis for future research:

- This model assumes that all parameters are deterministic and known. In real-world conditions, however, customer demand can fluctuate suddenly. Furthermore, the model assumes that machines do not break down and do not produce defective items, ensuring that production quantities meet customer demand on schedule. A breakdown could disrupt the fulfillment of customer orders. Product deliveries to customers are based on a fixed duration per shipment, and the time required to pickup parts from suppliers depends on the distance of each route. Traffic congestion can cause both deliveries and collections to fall behind schedule. Consequently, if this model is applied in a high-uncertainty environment, the resulting schedule may prove infeasible.
- The model assumes that deliveries are carried out sequentially by a single vehicle. While this assumption simplifies the problem, it limits applicability to real-world scenarios, specifically for companies with multi-vehicle fleets where different trips could potentially be executed in parallel to reduce the total time required to complete product deliveries to customers.
- Vehicles, whether for pickup or delivery, are assumed to be constantly operational. In reality, however,

vehicles may break down, and this would affect punctuality and the completion of all integrated activities

5.2.2. Suggestions for Future Research

Based on the outlined limitations, future research could consider stochastic parameters to account for uncertainty in real-world conditions. Additionally, the model could be extended to accommodate multiple delivery vehicles.

Acknowledgments

The authors gratefully acknowledge the financial support provided by the Indonesian Education Scholarship, the Center for Higher Education Funding and Assessment, and the Indonesian Endowment Fund for Education for this research under contract number: 01692/BPPT/BPI.06/9/2023.

Conflict of Interest

The authors declare no conflict of interest regarding the publication of this paper.

Declaration of AI Use

Throughout this research, the authors utilized Grammarly and Claude. Grammarly was used to enhance the quality and readability of the work, while Claude assisted in writing the Python scripts required to execute the computations for the designed algorithms. After using these tools, the author reviewed and adjusted the work to align with the specific problem addressed, and the author bears full responsibility for the content of this research.

References

- [1] M. Ben-Daya, R. As'Ad, and M. Seliaman, "An integrated production inventory model with raw material replenishment considerations in a three layer supply chain," *Int. J. Prod. Econ.*, vol. 143, no. 1, pp. 53–61, 2013, doi: 10.1016/j.ijpe.2010.10.024.
- [2] S. T. Zhao, K. Wu, and X. M. Yuan, "Optimal integer-ratio inventory coordination policy for an integrated multi-stage supply chain," *Appl. Math. Model.*, vol. 40, no. 5–6, pp. 3876–3894, Mar. 2016, doi: 10.1016/j.apm.2015.11.003.
- [3] M. Ben-Daya and A. Al-Nassar, "An integrated inventory production system in a three-layer supply chain," *Production Planning and Control*, vol. 19, no. 2, pp. 97–104, 2008, doi: 10.1080/09537280801895930.
- [4] W. Lee, "A joint economic lot size model for raw material ordering, manufacturing setup, and finished goods delivering," *Omega (Westport)*, vol. 33, no. 2, pp. 163–174, 2005, doi: 10.1016/j.omega.2004.03.013.
- [5] Z. L. Chen, "Integrated production and outbound distribution scheduling: Review and extensions," Jan. 2010. doi: 10.1287/opre.1080.0688.
- [6] S. Moons, K. Ramaekers, A. Caris, and Y. Arda, "Integrating production scheduling and vehicle routing decisions at the operational decision level: A review and discussion," Feb. 01, 2017, *Elsevier Ltd.* doi: 10.1016/j.cie.2016.12.010.

- [7] L. Berghman, Y. Kergosien, and J. C. Billaut, "A review on integrated scheduling and outbound vehicle routing problems," Nov. 16, 2023, *Elsevier B.V.* doi: 10.1016/j.ejor.2022.12.036.
- [8] E. Prasetyaningsih, Suprayogi, T. M. A. Ari Samadhi, and A. H. Halim, "Production and delivery batch scheduling with multiple due dates to minimize total cost," *Journal of Engineering and Technological Sciences*, vol. 49, no. 1, pp. 16–36, 2017, doi: 10.5614/j.eng.technol.sci.2017.49.1.2.
- [9] A. Agnetis, M. A. Aloulou, and M. Y. Kovalyov, "Integrated production scheduling and batch delivery with fixed departure times and inventory holding costs," *Int. J. Prod. Res.*, vol. 55, no. 20, pp. 6193–6206, Oct. 2017, doi: 10.1080/00207543.2017.1346323.
- [10] N. Karimi and H. Davoudpour, "Integrated production and delivery scheduling for multi-factory supply chain with stage-dependent inventory holding cost," *Computational and Applied Mathematics*, vol. 36, no. 4, pp. 1529–1544, Dec. 2017, doi: 10.1007/s40314-016-0305-0.
- [11] Y. Yin, Y. Yang, D. Wang, T. C. E. Cheng, and C. C. Wu, "Integrated production, inventory, and batch delivery scheduling with due date assignment and two competing agents," *Naval Research Logistics*, vol. 65, no. 5, pp. 393–409, Aug. 2018, doi: 10.1002/nav.21813.
- [12] D. Han, Y. Yang, D. Wang, T. C. E. Cheng, and Y. Yin, "Integrated production, inventory, and outbound distribution operations with fixed departure times in a three-stage supply chain," *Transp. Res. E Logist. Transp. Rev.*, vol. 125, pp. 334–347, May 2019, doi: 10.1016/j.tre.2019.03.014.
- [13] E. Yağmur and S. E. Kesen, "Integrated production scheduling and vehicle routing problem with energy efficient strategies: Mathematical formulation and metaheuristic algorithms," *Expert Syst. Appl.*, vol. 237, Mar. 2024, doi: 10.1016/j.eswa.2023.121586.
- [14] T. Azad, H. F. Rahman, R. K. Chakraborty, and M. J. Ryan, "Optimization of integrated production scheduling and vehicle routing problem with batch delivery to multiple customers in supply chain," *Memet. Comput.*, vol. 14, no. 3, pp. 355–376, Sep. 2022, doi: 10.1007/s12293-022-00372-x.
- [15] F. Marandi and S. M. T. Fatemi Ghomi, "Integrated multi-factory production and distribution scheduling applying vehicle routing approach," *Int. J. Prod. Res.*, vol. 57, no. 3, pp. 722–748, Feb. 2019, doi: 10.1080/00207543.2018.1481301.
- [16] S. Mohammadi, S. M. J. M. Al-e-Hashem, and Y. Rekik, "An integrated production scheduling and delivery route planning with multi-purpose machines: A case study from a furniture manufacturing company," *Int. J. Prod. Econ.*, vol. 219, pp. 347–359, Jan. 2020, doi: 10.1016/j.ijpe.2019.05.017.
- [17] H. Kuhn, D. Schubert, and A. Holzapfel, "Integrated order batching and vehicle routing operations in grocery retail – A General Adaptive Large Neighborhood Search algorithm," *Eur. J. Oper. Res.*, vol. 294, no. 3, pp. 1003–1021, Nov. 2021, doi: 10.1016/j.ejor.2020.03.075.
- [18] D. Hrabec, L. M. Hvattum, and A. Hoff, "The value of integrated planning for production, inventory, and routing decisions: A systematic review and meta-analysis," Jun. 01, 2022, *Elsevier B.V.* doi: 10.1016/j.ijpe.2022.108468.
- [19] P. Devapriya, W. Ferrell, and N. Geismar, "Integrated production and distribution scheduling with a perishable product," *Eur. J. Oper. Res.*, vol. 259, no. 3, pp. 906–916, Jun. 2017, doi: 10.1016/j.ejor.2016.09.019.
- [20] P. Farahani, M. Grunow, and H. O. Günther, "Integrated production and distribution planning for perishable food products," *Flex. Serv. Manuf. J.*, vol. 24, no. 1, pp. 28–51, Mar. 2012, doi: 10.1007/s10696-011-9125-0.
- [21] P. Lacomme, A. Moukrim, A. Quilliot, and M. Vinot, "Supply chain optimisation with both production and transportation integration: multiple vehicles for a single perishable product," *Int. J. Prod. Res.*, vol. 56, no. 12, pp. 4313–4336, Jun. 2018, doi: 10.1080/00207543.2018.1431416.
- [22] P. Maheshwari, S. Kamble, A. Belhadi, M. Venkatesh, and M. Z. Abedin, "Digital twin-driven real-time planning, monitoring, and controlling in food supply chains," *Technol. Forecast. Soc. Change*, vol. 195, Oct. 2023, doi: 10.1016/j.techfore.2023.122799.
- [23] C. N. Rosyidi, H. A. I. Permatasari, and P. W. Laksono, "An integrated optimization model for procurement and production lot sizing and scheduling problems," *Production Engineering Archives*, vol. 30, no. 2, pp. 155–165, Jun. 2024, doi: 10.30657/pea.2024.30.15.
- [24] S. Khanra, S. K. Ghosh, and C. Pathak, "A three-layer supply chain integrated production-inventory model with idle cost and batch shipment policy," *Sustainability Analytics and Modeling*, vol. 2, p. 100011, 2022, doi: 10.1016/j.samod.2022.100011.
- [25] M. Benbouja, A. Touil, A. Echchatbi, and A. Charkaoui, "Supply chain integration within mass customization: Tactical procurement, production and distribution modeling," *Journal of Industrial Engineering and Management*, vol. 14, no. 2, pp. 250–268, 2021, doi: 10.3926/jiem.3182.
- [26] N. Nikandish, K. Eshghi, and S. A. Torabi, "Integrated Procurement, Production and Delivery Scheduling in a Generalized three Stage Supply Chain," 2009.
- [27] A. S. Tibaldo, J. M. Montagna, and Y. Fumero, "A new integrated approach for solving batching, production scheduling and delivery problems in single-stage batch environments," *Comput. Chem. Eng.*, vol. 188, Sep. 2024, doi: 10.1016/j.compchemeng.2024.108753.
- [28] M. Dwi Astuti, Sukoyo, A. Cakravastia, and A. Hakim Halim, "An Integrated Optimization Model for Raw Material Pickup, Production, and Delivery to Minimize Total Relevant Cost." Presented at *Asia Pacific Conference on Manufacturing System (APCOMS)* and *International Manufacturing Engineering Conference (IMEC)* 2026."
- [29] A. H. Halim, S. Miyazaki, and H. Ohta, "Theory and Methodology Batch-scheduling problems to minimize actual flow times of parts through the shop under JIT environment," 1994.

- [30] Dale Varberg Edwin J. Purcell and Steve E. Rigdon, *Calculus Early Transcendentals*. Pearson Education Limited, 2014.

in press