JJMIE

Volume 19, Number 3, Sep. 2025
ISSN 1995-6665
Pages 669– 685
https://doi.org/10.59038/jjmie/190318
Editor-in-Chief: Prof. Ali M. Jawarneh

Jordan Journal of Mechanical and Industrial Engineering

Metaheuristic Approaches for Flexible Lot-sizing and Scheduling with Remanufacturing and Sequence-dependent Setup Time

Arezoo Osati[a], Esmaeil Mehdizadeh[a,*], Sadoullah Ebrahimnejad[b]

[a]*Department of Industrial Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran*
[b]*Department of Industrial Engineering, Karaj Branch, Islamic Azad University, Karaj, Iran.*

Received 29 Sep 2024 *Accepted 3 Jun 2025*

Abstract

This study addresses a critical research gap in production planning by proposing an integrated model for flexible lot-sizing and scheduling that simultaneously considers remanufacturing, sequence-dependent setup times, and energy efficiency—factors often studied in isolation in previous research. The model reflects the complexity of modern manufacturing systems, where products may return for remanufacturing and machines require specific setup times depending on operation sequences. The novelty of this work lies in its holistic approach, combining these elements within a flexible job-shop environment, which better captures the dynamics and constraints of real-world production settings. To efficiently solve this NP-hard problem, the study develops and applies three metaheuristic algorithms: Genetic Algorithm (GA), Whale Optimization Algorithm (WOA), and Particle Swarm Optimization (PSO). These algorithms are evaluated on a range of problem sizes to assess their scalability and performance. The key contribution is twofold: first, in the formulation of a realistic and comprehensive mathematical model, and second, in demonstrating the effectiveness of metaheuristic methods for solving complex large-scale problems. Results indicate that while all three algorithms produce feasible and high-quality solutions, the GA consistently achieves superior outcomes, making it a robust and efficient approach for optimizing lot-sizing and scheduling decisions in sustainable and flexible manufacturing environments.

Keywords: Remanufacturing, Lot-sizing, Flexible Job Shop, WOA, PSO, GeneticAlgorithm.

1. Introduction

In present time, manufacturing industries are facing a very competitive, unpredictable, and ever changing environment, with growing complexity and high levels of customization.(Kumar et al., 2017). In modern manufacturing environments, these problems are further complicated by factors such as remanufacturing processes, sequence-dependent setup times, and energy consumption constraints. These elements are not only essential for enhancing productivity and competitiveness but also align with the broader goals of sustainable and circular manufacturing systems (Jabbarzadeh et al., 2018).

The Flexible Job Shop Scheduling Problem (FJSP) extends the classical Job Shop problem by allowing each operation to be performed on more than one machine, offering greater routing flexibility. However, this increased flexibility introduces additional layers of computational complexity. The problem becomes even more challenging when integrated with lot-sizing decisions and the consideration of remanufacturing and setup dependencies (Giglio et al., 2017). Despite growing interest, most existing studies address these components separately. For instance, literature on remanufacturing primarily focuses on inventory or supply chain levels (Liu et al., 2025), while research on FJSP often ignores setup times and sustainability-related factors like energy consumption (Xin et al., 2025).

Research gaps emerge from this lack of integration. Specifically, few models simultaneously address lot-sizing and scheduling with remanufacturing, sequence-dependent setup times, and energy efficiency in a flexible job-shop environment (Xu et al., 2025). This fragmentation limits the applicability of existing models in real-world, resource-constrained, and sustainability-aware manufacturing systems. Therefore, this study aims to develop a unified, scalable, and computationally efficient model be developed for the integrated lot-sizing and scheduling problem that includes remanufacturing, sequence-dependent setup, and energy efficiency considerations.

To address this issue, the paper presents a comprehensive mathematical formulation of the problem, integrating all these critical elements into a single model. The novelty of the research lies in this integrated approach, as well as in the development and application of metaheuristic algorithms capable of solving the resulting NP-hard problem efficiently. In contrast to exact optimization methods that struggle with large-scale

* Corresponding author e-mail: emehdi@iau.ac.ir, emehdi04@gmail.com.

instances, metaheuristics provide near-optimal solutions within reasonable computational times.

The main contributions of this study are as follows:

1. Development of an integrated lot-sizing and flexible job-shop scheduling model incorporating remanufacturing, sequence-dependent setup times, and energy efficiency.
2. Implementation and comparative evaluation of three metaheuristic algorithms—Genetic Algorithm (GA), Whale Optimization Algorithm (WOA), and Particle Swarm Optimization (PSO)—to solve large-scale instances of the problem.
3. Empirical evidence showing the superiority of the GA in terms of solution quality and robustness, highlighting its potential for application in complex, real-world production systems.

By addressing these challenges, the study offers practical insights for sustainable production planning and advances the methodological frontier of integrated manufacturing optimization.

Figure 1 represents a simplified schematic view. The production system discussed here handles multiple product classes, each with unique properties and a specific precedence network. Each product class follows a single processing path across different workstations, with at least two similar machines performing each step in parallel. The processing path remains consistent for both production and remanufacturing methods, with fixed and predetermined nominal processing times on different machines. The allocated time interval for each operation has a predetermined duration. Metaheuristicsare effective methods for solving complex scheduling problems. They are iterative and stochastic procedures that can search for good solutions.

The main research gaps addressed by this study stem from the lack of integrated approaches that simultaneously consider lot-sizing and scheduling in the presence of remanufacturing, sequence-dependent setup times, and energy efficiency. While prior studies have individually addressed components such as flexible job-shop scheduling or remanufacturing, few have combined all these critical aspects into a unified model. Moreover, traditional exact optimization tools struggle to solve large-scale instances of such complex models within reasonable computation times, leaving a performance and scalability gap for practical applications. This paper specifically targets this void by constructing a comprehensive model that integrates all these real-world complexities into the flexible job-shop environment.

To tackle the computational challenges posed by the problem's NP-hard nature, the authors propose and implement three metaheuristic algorithms—Genetic Algorithm (GA), Whale Optimization Algorithm (WOA), and Particle Swarm Optimization (PSO). The key contribution lies not only in formulating the problem but also in evaluating the performance of these algorithms in large-scale scenarios, where exact solvers fail. The study demonstrates that the GA outperforms WOA and PSO in most tested scenarios, particularly in finding higher-quality solutions for large problem sizes. These findings contribute to both the methodological advancement in production planning optimization and the practical deployment of metaheuristic algorithms in modern, sustainable manufacturing systems that involve remanufacturing processes.

This study centers on the Sequence-Dependent FJSP (SDFJSP) and introduces meta-heuristic methods to address it. The research aims to create effective algorithms to yield good solutions for the FJSP, offering a useful understanding of meta-heuristic methods' effectiveness in tackling scheduling challenges in manufacturing. The paper continues with a literature review, mathematical modeling and problem formulation, and experimental results, followed by the conclusion and future research directions.

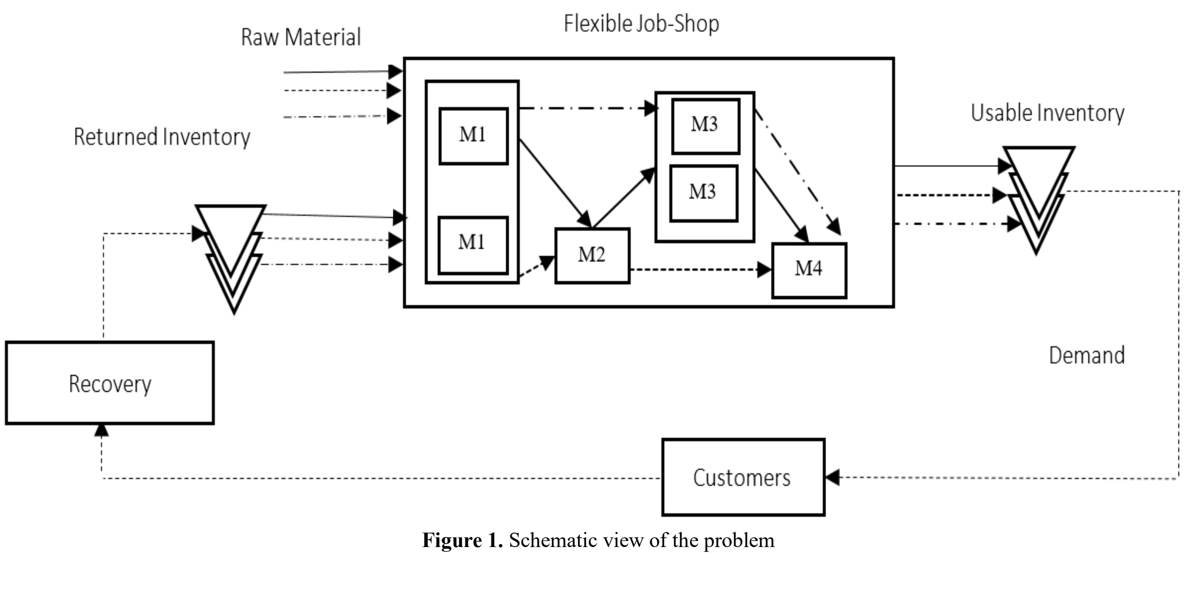

Figure 1. Schematic view of the problem

2. Previous Research

2.1. Flexible Job-shop Scheduling Problem (FJSP)

The FJSP encompasses versatile machinery for various operations. Researchers have developed numerous solution methods. Brucker and Schlie (1990) offered an efficient algorithm. Many others used Tabu search (Brandimarte, 1993; Hurink et al.,1994), mixed integer programming (Choi and Choi, 2002), Genetic Algorithm(Kacem et al., 2002; Chan et al., 2006; Gao et al.,2008; Pezzella et al., 2008). Later, approaches like decision support systems, hybrid algorithms, and multi-agent systems have been utilized. Industrial applications have been studied, and adaptive strategies for set-up times, batch processing, and machine failure have been proposed to optimize scheduling efficiency (Xing et al., 2010; Chen et al., 2008; Zhang et al., 2011; Mahdavi et al. 2010; Chan and Choy, 2011; Al-Turki et al., 2011; Xiong and Fu, 2018; Gao et al., 2015; Ahmadi et al., 2016).

In recent research, Mahmud et al. (2022) present a scheduling problem that addresses the challenges of achieving customized, on-time deliveries while minimizing costs. Their approach combines a supply portfolio with production planning by utilizing a flexible job shop (FJS) model to represent manufacturing adaptability.They develop CD-MOPSO and RP-MOPSO methods to enhance multi-objective PSO, with RP-MOPSO outperforming by providing diverse Pareto solutions efficiently. In a related context, Thi et al. (2022) propose an improved algorithm for FJSP under the failure of machines, utilizing GA for initial solutions during failures. Meanwhile, Momenikorbekandi and Abbod (2023) introduce a novel metaheuristic for FJSP. The algorithm combines genetic and parthenogenetic algorithms, incorporating ethnic selection and various selection operators, demonstrating better performance for a range of scheduling scenarios. These studies collectively contribute to the optimization of complex scheduling problems through innovative metaheuristic approaches and practical adaptability.

Shen et al (2023) addresses energy-efficient scheduling in flexible job shops under time-of-use electricity pricing, aiming to minimize total energy cost without exceeding a maximum makespan. They develop heuristics and a tabu search based on problem properties, showing how energy pricing structures impact solution complexity and cost savings. Fontes et al (2024) extends energy-efficient job shop scheduling by incorporating speed-adjustable machines and transport vehicles, optimizing both processing and transport tasks. A bi-objective MILP model and a multi-objective genetic algorithm are proposed to find Pareto-optimal trade-offs between makespan and energy consumption, with results demonstrating efficiency even for large instances. Adams (2025) develops a mixed-integer linear production planning model for energy-intensive manufacturing using hybrid battery–hydrogen storage systems. Numerical results show up to 29.3% cost savings, with batteries reducing energy costs and hydrogen storage enhancing energy independence under fluctuating renewable supply and tariffs.

2.2. Lot-Sizing with remanufacturing

Recently, there has been an increasing focus on improving production systems, specifically concerning lot-sizing and scheduling. Efforts are directed towards creating better solutions through metaheuristic algorithms like genetic algorithms and simulated annealing, alongside decision support systems and dispatching rules.

The FJSP, which entails the assignment of various products to parallel machines considering setup times, stands out as a significant area of interest. Various strategies, such as genetic algorithms and decision support systems, have been suggested to tackle this problem, but its inherent complexity and the requirement to harmonize competing goals continue to make it a difficult problem to solve.

The DLSPR problem (which stands for Dynamic Lot Sizing with Product Returns and Remanufacturing) focuses on determining the optimal quantities for the production of both new and remanufactured items, considering the fluctuating and uncertain nature of product returns.

Overall, the optimization of production systems is a complex and challenging problem that requires a multidisciplinary approach. By combining mathematical modeling, metaheuristic algorithms, and decision support systems, researchers can develop more effective methods for solving these problems and improving the efficiency of production systems.

Over the past five decades, the DLSPR has been reviewed by many researchers (Drexl and Kimms, 1997; Jans and Degraeve, 2008). Richter and Sombrutzki (2000) as well as Golany et al. (2001) have contributed to this field by providing mathematical formulations for DLSP in remanufacturing and analyzing single product systems with reproduction and disposal options, respectively. Teunter et al. (2006) analyzed capacity planning without capacity constraints for joint and separate production and remanufacturing lines.

Teunter et al. (2009) formulated heuristic approaches for problems initially introduced by Teunter in 2006. Pineyro and Viera (2009) developed a Tabu Search algorithm aimed at solving a DLSR problem that encompasses sustainableobjectives. Additionally, Pineyro and Viera. (2010) examined a lot-sizing problem where the demand for new products does not match the demand for remanufactured ones, resulting in a situation of one-way substitution, and they offered a Tabu Search technique to find a solution that is close to optimal.Wang et al. (2011) considered separate production lines for production and remanufacturing, allowing outsourcing to meet demand. Baki et al (2014)., Sifaleras et al. (2015), and Mehdizadeh and Fatehi (2017) proposed various algorithms for DLSPR and recycling without assuming capacity constraints.

Soleimaninia and Mehdizadeh (2018) developed a model for FJSPand a GA to solve it.Aghighi et al. (2021)a sustainable JSP model without taking into account the processing dependencies.

Van Zyl and Adetunji (2022) presented a lot-sizing problem involving imperfect manufacturing processes, time-varying demand, and return rates. Recognizing the importance of return logistics in the face of natural resource scarcity, the study proposes an inventory system that integrates remanufactured returns with new products, optimizing value creation throughout the product life cycle. The authors devise a sophisticated approach, including the consideration of returns that cannot be fully remanufactured, along with failed manufacturing items treated as returns. Remanufactured items are classified into different quality grades, aligning them with diverse customer demands. The study employs a modified Wagner/Whitin model to efficiently balance remanufacturing and manufacturing activities while minimizing costs, showing sensitivity to manufacturing setup costs and return proportions.

In a parallel vein, Rocha (2023) introduces a novel model for the capacitated lot-sizing problem with remanufacturing (CLSP-RM). The study explores two distinct instance groups: those amenable to relaxation-based solutions yielding near-optimal outcomes and those categorized as NP-hard, solved through straightforward period-by-period simulation techniques. By offering an innovative model formulation that considers setup costs, product returns, and remanufacturing, Rocha contributes to the advancement of solution approaches for the complex CLSP-RM, providing valuable insights into efficient optimization strategies for manufacturing and remanufacturing scenarios.

Mat Ropi et al (2023) develops a cost-minimization model for managing disruptions in remanufacturing systems caused by spare parts shortages, focusing on optimal recovery scheduling in a two-stage production–inventory setup. Rohaninejad et al (2024) addresses integrated lot-sizing and scheduling using Reconfigurable Machine Tools (RMTs) under workforce constraints, formulating the problem as a MILP model. A decomposition heuristic combining MILP and constraint programming is proposed, demonstrating improved performance and highlighting the operational benefits of RMTs through key performance indicators. Vidal et al (2025) addresses lot-sizing and transportation planning in a hybrid production–remanufacturing system with recovery targets, inventory costs, and setup costs. A mixed-integer linear model and Tabu Search-based heuristics are developed, with results showing the heuristics outperform exact methods on large instances in both cost and computational time.

2.3. The problem of Integrated lot-sizing and scheduling

Rohaninejad et al. (2015) introduced models and hybrid optimization techniques to solve lot-sizing and FJSP. In addition, Rohaninejad et al. (2016) investigated both problems using a hybrid metaheuristic method that merges genetic algorithms with PSO. Sahraeian et al. (2017) approached the integration of lot-sizing with job-shop scheduling by applying harmony search along with mixed integer programming. The FJSP is a scheduling challenge that assigns various jobs to machines. The sequence-dependent FJSP (SD-FJSP) adds constraint to the problem for sequences (Xie et al., 2019).

Several studies have addressed the SD-FJSP in recent years, with a focus on developing effective optimization algorithms that can solve the problem efficiently. For example, some researchers have proposed using GA, PSO, and other metaheuristic techniques (Jiang et al., 2022).

One of the key challenges in solving the SD-FJSP is the large search space resulting from the sequencing constraint. In addition, the problem is NP-hard (Xie et al., 2019).

Despite these challenges, the SD-FJSP has important applications in manufacturing and other industries, where efficient scheduling can lead to significant cost savings and productivity improvements. As a result, there is ongoing research aimed at developing new algorithms and techniques for solving the problem, as well as improving the accuracy and realism of the models used to represent real-world scenarios.

Zarrouk et al. (2019) and Zhang et al. (2020) present a two-stage PSO to solve FJSP with a boundary-checking strategy for efficiency in reducing scheduling decision variables in flexible jobshops. Hajibabaei and Behnamian (2021) examine flexible resources impact on scheduling with parallel machines, using linear models and comparing with genetic algorithms aided by Tabu Search for larger instances. Osati et al. (2022) tackle integrated lot-sizing and FJSP, emphasizing energy efficiency, through a MINLP model solved by exact methods.

Carvalho & Nascimento (2022) addresses integrated lot-sizing and scheduling problem on parallel non-identical machines with complex setup constraints, aiming to minimize production, setup, and inventory costs. The results show that novel matheuristic approaches combining relax-and-fix, fix-and-optimize, path-relinking, and kernel search outperform CPLEX in solving industrial-scale instances efficiently.

Rosyidi et al (2024) developed an integrated model for simultaneous procurement and production lot sizing and scheduling, considering multiple suppliers with quantity discounts to maximize profit. A real-world case in the noodle industry demonstrates the model's effectiveness, revealing sensitivity of results to procurement and pricing parameters.

Kumar et al (2025) presents a hybrid optimization model combining stochastic programming and MILP to solve the Lot Sizing and Scheduling Problem under demand uncertainty. The model enhances cost-effectiveness and robustness, offering improved production planning in uncertain manufacturing environments.

Rohaninejad et al (2025) addresses a multi-product lot-sizing and scheduling problem with a novel period-based learning effect, formulating a MILP model to minimize tardiness and overtime costs. Cutting planes, matheuristics, and post-processing improve solution quality, with results highlighting the learning effect's significant influence on performance and sensitivity to time-based parameters.

2.4. Research Gaps and Contributions:

Despite extensive studies on lot-sizing, scheduling, and remanufacturing, a significant research gap remains in the integrated treatment of these aspects within flexible job-shop environments that include sequence-dependent setup times and energy efficiency considerations. Prior works often handle these elements in isolation, limiting their real-world applicability, especially in sustainable and resource-constrained production systems. This study fills that gap by proposing a unified mathematical model that integrates lot-sizing and sequence-dependent flexible job-shop scheduling with remanufacturing and energy efficiency objectives. To overcome the NP-hard nature of the problem, we implement and compare three metaheuristic algorithms—Genetic Algorithm (GA), Whale Optimization Algorithm (WOA), and Particle Swarm Optimization (PSO). The research contributes novel insights by demonstrating the GA's superiority in solution quality for large-scale scenarios and offering a practical, scalable optimization framework that advances both theory and practice in sustainable production planning.More specifically, the research questions include:

1. How can the problem of Flexible Lot-sizing and Scheduling with Remanufacturing and Sequence-dependent Setup Time be modeled in an integrated manner?
2. How can the problem of Flexible Lot-sizing and Scheduling with Remanufacturing and Sequence-dependent Setup Time be solved using metaheuristic algorithms?

3. Problem formulation

This section outlines model variables, constraints, and objectives as a mathematical model which is a mixed integer non-linear model.

3.1. Sets

Classes of products: $\mathcal{P} = \{1. \dots . P\}$
Machines: $\mathcal{K} = \{1. \dots . K\}$
Periods: $\mathcal{T} = \{1. \dots . T\}$
Production methods: $\mathcal{F} = \{M. R\}$; including manufacturing (M) and remanufacturing (R)
Operations: $\mathcal{H}_i = \{1. \dots . H_i\}$, $i \in P$
Processing Machines: $\mathcal{O}_i = \left\{K(1). K(2). \dots . K\left(o_{iH_i}\right)\right\}$, where $K(o_{ih})$denotes the machines that can operateo_{ih} ($h \in \mathcal{H}_i$).

3.2. Parameters

H_i: Total operations of product $i \in P$
q_{ihk}^f: Normal time needed for$h \in \mathcal{H}_i$, $i \in P$,$k \in K(o_{ih})$, $f \in F$
e_{ihk}^f: Allowable compression of $h \in \mathcal{H}_i$, $i \in P$, $k \in K(o_{ih})$, $f \in F$
r_{it}: Quantity ofproducts returned, $i \in P$,$t \in T$
d_{it}: Demand, $i \in P$, $t \in T$
v_{it}^f: Production cost, $i \in P$, $t \in T$, $f \in F$
$w_{ihi'h'}^k$: Setup cost, $k \in K(o_{ih}) \cap K(o_{i'h'})$, the operation $h'$of$i' \in P$is executed afterthe operation$h$of $i \in P$)
s_{ih}^k: Setup time,$k \in K(o_{ih}) \cap K(o_{i'h'})$,$h \in \mathcal{H}_i$, $i \in P$
$p_{hh'}^i$: Precedence parameter, $p_{hh'}^i$=1 if $h$is a predecessorof$h' \in \mathcal{H}_i$, $i \in P$
h_{it}^u: Holding cost (applied for returned items),$i \in P$, $t \in T$
h_{it}^s: Holding cost (applied for normal items), $i \in P$, $t \in T$
h_{it}^b: Cost of shortage, $i \in P$, $t \in T$
c_t: Total length of period,$t \in T$
cp_k: Energy consumption cost,$k \in K$
ci_k: Power consumption cost,$k \in K$
cc_k: Compression cost,$k \in K$
ψ: A sufficiently large value
b_{it}^f: Production capacity, $i \in P$, $t \in T$,$f \in F$

3.3. Decision variables

3.3.1. Integer variables

$X_{it}^f \geq 0$: Production quantity,$i \in P$, $t \in T$, $f \in F$
$I_{it}^u \geq 0$: Inventory of products returned,$i \in P$, $t \in T$
$I_{it}^s \geq 0$: Inventory of the normal products,$i \in P$, $t \in T$
$I_{it}^b \geq 0$: Shortage, $i \in P$,$t \in T$

3.3.2. Continuous variables

C_{PHS}: Total cost (cost of production, inventory cost, and setup cost)
C_E: Energy cost
$PT_{ihkt}^f \geq 0$: Realized processing time, $h \in \mathcal{H}_i$,$i \in P$, $k \in K(o_{ih})$, $t \in T$, $f \in F$.
$Z_{ihkt}^f \geq 0$: The amount of compression,$h \in \mathcal{H}_i$,$i \in P$, $k \in K(o_{ih})$, $t \in T$, $f \in F$.
$ST_{ihkt}^f \geq 0$: Execution start time, $h \in \mathcal{H}_i$, $i \in P$, $k \in K(o_{ih})$, $t \in T$, $f \in F$.
CO_{kt}: Last active time of machine$k \in K$, $t \in T$
IT_{kt}: Idle time of machine $k \in K$, $t \in T$.

3.3.3. Binary variables

$\delta_{ihi'h'}^{ktfg} \in \{0,1\}$: Equal to 1 if operations $h'$of the product i'in t and production method g is scheduled on the machine k after operations h of the product i with production method f. Otherwise $\delta_{ihi'h'}^{ktfg}$=0.

$y_{ihkt}^f \in \{0,1\}$: Equal to 1 if operations h of the product $i \in P$ is scheduled t on the machine k using method $f \in F$; Otherwise y_{ihkt}^f=0.

B_{it}^f: Equal to 1 if anon-zero quantity of product i in $t$is produced using the method $f \in F$. Otherwise B_{it}^f=0.

3.4. Mathematical model

$$Min\ \mathcal{C}_{PHS} + \mathcal{C}_E \tag{1}$$

St:

$$\mathcal{C}_{PHS} = \sum_{t\in T}\sum_{i\in P}\sum_{f\in \mathcal{F}} v_{it}^f X_{it}^f + \sum_{t\in T}\sum_{i\in P}\sum_{i'\in P}\sum_{h\in H_i}\sum_{h'\in H_{i'}}\sum_{k\in K}\sum_{f\in \mathcal{F}}\sum_{g\in \mathcal{F}} \delta_{ihi'h'}^{ktfg}\, w_{ihi'h'}^k + \sum_{t\in T}\sum_{i\in P}(h_{it}^u I_{it}^u + h_{it}^s I_{it}^s + h_{it}^b I_{it}^b) \tag{2}$$

$$C_E = \sum_{t\in T}\sum_{i\in P}\sum_{k\in K(o_{ih})}\sum_{h\in H_i}\sum_{f\in \mathcal{F}} cp_k PT^f_{ihkt} + \sum_{t\in T}\sum_{k\in \mathcal{K}} ci_k IT_{kt} + \sum_{t\in T}\sum_{k\in \mathcal{K}}\sum_{i\in P:k\in K(o_{ih})}\sum_{f\in \mathcal{F}}\sum_{h\in H_i} c_k Z^f_{ihkt} \tag{3}$$

$$IT_{kt} = CO_{kt} - \sum_{i\in P:k\in K(o_{ih})}\sum_{f\in \mathcal{F}}\sum_{h\in H_i} PT^f_{ihkt} \quad \forall t\in \mathcal{T}.\forall k\in \mathcal{K} \tag{4}$$

$$I^u_{it} - I^u_{i(t-1)} + X^R_{it} = r_{it} \quad \forall t\in \mathcal{T}.\forall i\in \mathcal{P} \tag{5}$$

$$I^s_{i(t-1)} - I^s_{it} + \sum_{f\in \mathcal{F}} X^f_{it} + I^b_{it} - I^b_{i(t-1)} = d_{it} \quad \forall t\in \mathcal{T}.\forall i\in \mathcal{P} \tag{6}$$

$$X^f_{it} \le b^f_{it} B^f_{it} \quad \forall t\in \mathcal{T}.\forall i\in \mathcal{P}.\forall f\in \mathcal{F} \tag{7}$$

$$\sum_{h\in \mathcal{H}_i}\sum_{k\in \mathcal{K}} y^f_{ihkt} = |\mathcal{H}_i| B^f_{it} \quad \forall t\in \mathcal{T}.\forall i\in \mathcal{P}.\forall f\in \mathcal{F} \tag{8}$$

$$\sum_{k\in K(o_{ih})} y^f_{ihkt} \le 1 \quad \forall t\in \mathcal{T}.\forall i\in \mathcal{P}.\forall h\in \mathcal{H}_i.\forall f\in \mathcal{F} \tag{9}$$

$$ST^f_{ihkt} \le y^f_{ihkt} c_t \tag{10}$$

$$PT^f_{ihkt} \le y^f_{ihkt} c_t \tag{11}$$

$$PT^f_{ihkt} = q^f_{ihk} X^f_{it} - Z^f_{ihkt} + y^f_{ihkt} s^k_{ih} \quad \forall t\in \mathcal{T}.\forall k\in \mathcal{K}.\forall i\in \mathcal{P}: \mathcal{O}_i \cap \{k\} \ne \emptyset.\forall f\in \mathcal{F}.\forall h\in \mathcal{H}_i \tag{12}$$

$$Z^f_{ihkt} \le e^f_{ihk} y^f_{ihkt} \quad \forall t\in \mathcal{T}.\forall k\in \mathcal{K}.\forall i\in \mathcal{P}: \mathcal{O}_i \cap \{k\} \ne \emptyset.\forall f\in \mathcal{F}.\forall h\in \mathcal{H}_i \tag{13}$$

$$CO_{kt} \le c_t \quad \forall t\in \mathcal{T}.\forall k\in \mathcal{K} \tag{14}$$

$$ST^f_{ihkt} + PT^f_{ihkt} \le CO_{kt} \quad \forall t\in \mathcal{T}.\forall k\in \mathcal{K}.\forall i\in \mathcal{P}: \mathcal{O}_i \cap \{k\} \ne \emptyset.\forall f\in \mathcal{F} \tag{15}$$

$$p^i_{hh'}\left(\sum_{k\in \mathcal{K}} ST^f_{ihkt} + PT^f_{ihkt}\right) \le \sum_{k\in \mathcal{K}} ST^f_{ih'kt} \quad \forall t\in \mathcal{T}.\forall i\in \mathcal{P}.\forall h.h'\in \mathcal{H}_i.\forall f\in \mathcal{F} \tag{16}$$

$$\psi\left(1-\delta^{ktfg}_{ihi'h'}\right) + ST^f_{i'h'kt} - ST^g_{ihkt} \ge PT^g_{ihkt} \quad \forall t\in \mathcal{T}.\forall k\in \mathcal{K}.\forall f.g\in \mathcal{F}.\forall i.i'\in \mathcal{P}: \mathcal{O}_i \cap \mathcal{O}_{i'} \cap \{k\} \ne \emptyset.\left((i \ge i') \wedge (f \ne g)\right) \vee \left((i > i') \wedge (f = g)\right).\forall h.h' \in \bigcup_{i=1}^{\mathcal{P}} \mathcal{H}_i \tag{17}$$

$$\psi\delta^{ktfg}_{ihi'h'} + ST^g_{ihkt} - ST^f_{i'h'kt} \ge PT^f_{i'h'kt} \quad \forall t\in \mathcal{T}.\forall k\in \mathcal{K}.\forall f.g\in \mathcal{F}.\forall i.i'\in \mathcal{P}: \mathcal{O}_i \cap \mathcal{O}_{i'} \cap \{k\} \ne \emptyset.((i \ge i') \wedge (f \ne g)) \vee ((i > i') \wedge (f = g)).\forall h.h' \in \bigcup_{i=1}^{\mathcal{P}} \mathcal{H}_i \tag{18}$$

The presented model defines the objective function in equation (1) which includes production and energy costs. The production cost is broken down into production, sequence-dependent setup, inventory, and remanufacturing costs as shown in constraint (2). In addition, constraint (3) represents the energy cost. The calculation of idle times of machines is expressed in constraint (4), that is defined as the difference between available time and completion time at each time period. The constraints (5) and (6) establish the formulas governing the changes in returned and available-to-deliver inventories. Note that back-ordered inventory is not allowed for returned products. However, equation (6) handles back-order in available-to-deliver inventory. Finally, the constraint (7) states that$B^f_{it} = 1$if$X^f_{it} > 0$.

Constraint (8) ensures that required production tasks only can occur within the same time period ($B^f_{it} = 1$). The constraint (9) stipulates that each task can only be assigned to one machine within a given time period. The constraints (10) and (11) guarantee that both the start time and duration of the task y^f_{ihkt} do not exceed the maximum time allocated for the underlying period. Constraint (12) calculates a compressed processing time on a specific machine k, whereas constraint (13)addresses the maximum allowable compression time (which is greater than 0 only if some tasks are assigned to that machine). The constraints(14) and (15) mandates all machines to complete their tasks within the time provided. The constraint(16) ensures that the precedence relation between to subsequent tasks is satisfied. Lastly, constraints(17) and (18) are used to assign the sequence variables $\delta^{ktfg}_{ihi'h'}$.A big bucket model is used to handle sequence decisions with variables $\delta^{ktfg}_{ihi'h'}$. According to the constraints (17) and (18), the variables $\delta^{ktfg}_{ihi'h'}$ can take value of 1 if and only if the tasks $i'h'$ and ih are immediately implemented in same mode and on same machine.

3.5. *Metaheuristic Approaches*

3.5.1. *Genetic Algorithm*

The GA, conceptualized by Holland (1992), is a search strategy based on evolutionary concepts to find the best solutions to problems. It works with a group of potential solutions, symbolized as chromosomes, and assesses each one using a fitness function to determine its effectiveness (Katoch et al., 2021).

The GA first generates N individuals, each represented by a chromosome. The chromosomes are composed of genes, which are usually represented as binary strings, but can also be represented in other ways depending on the problem being optimized. The algorithm then evolves the population over several generations, each consisting of the following steps:

Selection: selecting a subset of individuals to be used as parents based on the fitness of each individual.

Crossover: The selected parents are recombined to produce offspring for the next generation. This is done by randomly selecting crossover points along the length of the chromosomes and exchanging the corresponding segments between the parents. This process creates new individuals that inherit traits from both parents.

Mutation: A small random mutation is introduced to the offspring chromosomes. This is done by randomly flipping some of the bits in the chromosome with a low probability.

Evaluation: This is done using the fitness function.

Replacement: Adding new individuals to the population, and least fit individuals are deleted from the generation.

The Genetic Algorithm iterates through cycles of selection, solution change, and random alterations, stopping only when it reaches a set number of generations or satisfies specific conditions like reaching a cap on iterations or an acceptable level of precision. The decision about which solutions are promising is informed by a fitness function, which evaluates their efficacy. The essence of this algorithm is outlined in a structured, step-wise pseudo-code format:

```
Initialize the population of chromosomes
Set maximum_generations
for generation in range(maximum_generations):
        Calculate fitness for each chromosome
        Select parents for reproduction (e.g., using a roulette
        wheel or tournament selection)
        Create new offspring through crossover and mutation
        Evaluate the fitness of offspring
        Select chromosomes for the next generation (e.g.,
        using elitism or replacement)
Return the best chromosome from the final generation
```

Pseudo-code1. The pseudo-code of GA

3.5.2. *Whale Optimization Algorithm (WOA)*

The WOA is a nature-inspired metaheuristic, designed to mimic the communal hunting patterns of humpback whales when they feed. The algorithm was introduced by Mirjalili (2016).

The WOA algorithm starts by initializing potential solutions. The algorithm evaluates each solution's fitness function, which measures how well a solution satisfies the optimization problem's objective. The fitness values are then used to determine the best solution (Gbest), and the best solution for each individual (Pbest).

The phases of the WOA (exploration and exploitation) are based on the distance between the current solution and the Gbest. When the distance is small, the algorithm focuses on exploitation by updating the solutions towards the Gbest. When the distance is large, the algorithm focuses on exploration by randomly generating new solutions. The exploration and exploitation phases are controlled by a parameter called a.The updated equations for the WOA algorithm are as follows:

1. Update the position of the search agents:

   $D = C * Gbest_{pos} - whale_{pos}$

   $new\_whale_{pose} = Gbest_{pos} - A * D$

   where $\mathrm{whale_{pos}}$ is the current position of the search agent, $\mathrm{Gbest_{pos}}$ is the global best position, C is a random vector between -1 and 1, and A is the search agent's search range.
2. Boundary checking:

   If any decision variable of a new solution goes beyond the search space's boundaries, it is reset to the boundary value.
3. Update the search range:

   A = 2 - iter * ((2) / max_iter)

   Where iter and max_iter arethe current iteration and is maximum number of iterations, respectively.
4. Update the Gbest and Pbest:

   If a recently generated solution surpassesGbest, the Gbest is substituted with that solution. Likewise, if it improves upon Pbest, the Pbest is replaced with it. The framework of the WOA is captured in the following pseudo-code format:

```
Initialize whale population
Initialize best_solution
Set maximum_iterations
for iteration in range(maximum_iterations):
for each whale in the population:
Calculate fitness for the current whale
if fitness is better than fitness of best_solution:
Update best_solution
    a = 2 - 2 * iteration / maximum_iterations  # Linearly
       decreasing a
    A = 2 * a * random() - a
    C = 2 * random()
for each dimension in the solution:
      D = abs(C * best_solution[dimension] - whale[dimension])
      new_solution[dimension] = best_solution[dimension] - A *
      D
Apply bounds to new_solution
if random() < 0.5:
      Update whale using new_solution
else:
      Randomly update the whale's position
Update a
Return best_solution
```

Pseudo-code 2. The pseudo-code of WOA

3.5.3. *Particle Swarm Optimization (PSO)*

The PSO algorithm(Eberhart and Kennedy, 1995), is an optimization method informed by the collective movement patterns of birds or fish. It utilizes a group of 'particles', each signifying a potential solution, which navigates the solution population, guided by both their knowledge and the collective insights of the entire group, in pursuit of the optimal solution. PSO is widely used for different problems;Rahimi and Fazlollahtabar successfully implemented hybrid particle swarm and genetic algorithms for closed-loop green supply chains(Rahimi et al., 2018).

The PSO algorithm initializes$N$ particles in a random way. The particle $i$is displayed by a vector $x_i =$

$(x_{i1}, x_{i2}, \dots, x_{id})$ and a velocity vector $v_i = (v_{i1}, v_{i2}, \dots, v_{id})$, ($d$: dimension of the solution vector). The algorithm aims to find the optimal values of $x_i$with the objective function $f(x)$.

The PSO uses the following equations:

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1)x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1)$$

$$v_{ij}(t+1) = w * v_{ij}(t) + c_1 * rand(0,1) * \left(p_{ij}(t) - x_{ij}(t)\right) + c_2 * rand(0,1) * \left(p_{gj}(t) - x_{ij}(t)\right) * v_{ij}(t+1) = w * v_{ij}(t) + c_1 * rand(0,1) * \left(p_{ij}(t) - x_{ij}(t)\right) + c_2 * rand(0,1) * \left(p_{gj}(t) - x_{ij}(t)\right)$$

The notation $x_{ij}(t)$ refers to the specific element of the i^{th} particle's position in the j^{th} dimension during the t^{th} iteration. Similarly, $v_{ij}(t)$ specifies the velocity for the same particle and dimension at that iteration. The term w denotes the inertia weight, influencing the momentum of particles. Cognitive and social behaviors are parameterized by c_1 and c_2, respectively, guiding particles towards their personal best ($p_{ij}(t)$) and the swarm's global best ($p_{gj}(t)$) positions. Random factors in the movement are introduced by (0,1), a uniform random number.

Position updates are calculated by adding the velocity component, which in turn is updated based on past velocity, and individual and collective experiences, moderated by cognitive and social parameters. The inertia weight w plays a pivotal role in dictating strategic behavior: high values lead to extensive exploration across the potential solution landscape, while low values prompt a quicker, more focused convergence on the best-known solution.

The PSO algorithm terminates when a stopping criterion is met, such as a maximum number of iterations or a minimum error threshold.The pseudo-code of PSO is as follows:

```
Initialize particle positions and velocities
Initialize best_particle_position and global_best_position
Set maximum_iterations
for iteration in range(maximum_iterations):
for each particle:
Calculate fitness for the current particle
If fitness is better than fitness of best_particle_position:
        Update best_particle_position
If fitness is better than the fitness of global_best_position:
        Update global_best_position
Update particle velocity
For each dimension in velocity:
        Apply constraints to velocity components
Update particle position
Return global_best_position
```

Pseudo-code 3. The pseudo-code of PSO

4. Experimental Results

4.1. Numerical results

In this section, a comprehensive numerical example is presented to illustrate the problem framework and evaluate the performance of the proposed optimization model across various problem sizes. The model was implemented in GAMS using the BARON solver, with the objective of minimizing total operational costs, including both energy consumption and production expenses.

The illustrative case utilized sample input data provided in Tables 1 through 6. Upon processing these inputs, the model yielded an optimal solution, the results of which are detailed in Tables 7 to 9 and Figures 3 to 6. The obtained minimum total cost was 1424.

To evaluate the effectiveness of the proposed model, its results were compared with those of conventional batch-sizing approaches that consider only production costs. When energy costs were excluded from the objective function, the model achieved a lower objective value of 406, as illustrated by the Gantt charts in Figures 5 and 6.

A detailed comparison of both models is provided in Table 9. Notably, incorporating energy costs resulted in an overall cost reduction of 914 units and a substantial decrease in machine idle time, from 938 to 92 time units. These findings demonstrate that the integrated cost model is more efficient and economically advantageous than the traditional approach.

In conclusion, this example underscores the effectiveness of the proposed model in minimizing total costs through the inclusion of energy consumption, a critical factor often neglected in conventional production planning models. By explicitly incorporating energy-related expenses into the objective function, the model facilitates significant cost savings and enhances overall production efficiency.

Table 1. Illustrative example

I	T	M	H_1	H_2
2	2	3	4	5

Table 2. Allowable machines for different activities

i	Activity	Machine		
		k=1	k=2	k=3
i=1	h=1	1	0	1
i=1	h=2	1	1	0
i=1	h=3	1	0	1
i=1	h=4	1	1	1
i=2	h=1	0	1	1
i=2	h=2	0	1	0
i=2	h=3	1	0	1
i=2	h=4	1	1	0
i=2	h=5	0	0	1

Table 3. Returned products

Product \ time interval	t=1	t=2
i=1	10	14
i=2	8	10

Table 4. Demand values

Product \ time interval	t=1	t=2
i=1	15	16
i=2	12	18

Table 5. Per-product cost of production

i	t	Production method	
		M	R
i=1	t=1	4	2
i=1	t=2	6	4
i=2	t=1	5	3
i=2	t=2	8	5

Table 6. Capacity of time intervals

Interval	t=1	t=2
Capacity	250	300

Precedence networks for both the product classes are presented in the figure 2:

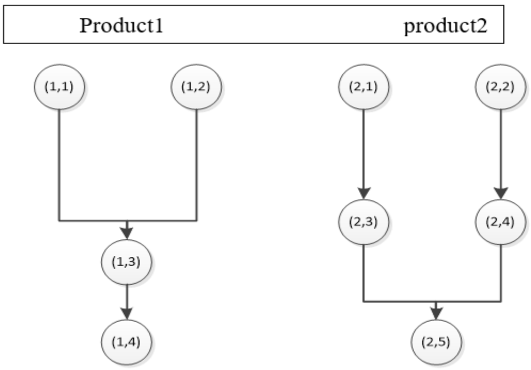

Figure 2. Precedence networks for both the product classes

4.1.1. Problem with energy efficiency considerations

Table 7 details the levels of production over various time intervals with an emphasis on energy efficiency. An analysis of the information within this table reveals that during the second time interval, manufacturing activity was absent for products belonging to class 2. This observation might be indicative of strategic scheduling that factors in energy efficiency or other operational constraints impacting the production timeline for these specific items.

Table 7. Production with Energy Efficiency Consideration

i	t	F	
		M	R
i =1	t=1	5	10
i =1	t=2	2	14
i =2	t=1	22	8

The Gantt charts presented in Figures 3 and 4 provide a comprehensive overview of the schedule of activities for each time interval. The Gantt charts are presented with great attention to detail, allowing for a clear understanding of the manufacturing and remanufacturing operations conducted in the system.

To facilitate the visual interpretation of the charts, the manufacturing operations are displayed as grey bars, while the remanufacturing operations are represented as green bars. By incorporating this color-coded system, the charts become highly readable and aid in the analysis of the system's overall performance.

Moreover, to identify activities, the numbers displayed on each bar are presented as ordered tuples (i.h.k.t.f), where each component of the tuple corresponds to a specific dimension of the activity. This approach ensures that every activity is uniquely identifiable and allows for a more detailed examination of the system's operations.

4.1.2. Problem with no energy efficiency consideration

Table 8 presents production volume across each time slot without the integration of energy efficiency considerations. The corresponding production timeline is visually mapped out through Gantt charts in Figures 5 and 6, which serve to chronologically demonstrate the scheduled operations within designated periods.

For clearer comprehension and distinction in the Gantt charts, a color-coding system is employed: manufacturing processes are marked with grey bars, and remanufacturing ones with green bars. This color distinction aids readers in quickly discerning the type of operation at a glance without confusion.

Moreover, to further facilitate the identification of each specific activity, a numerical coding system is incorporated into the charts. Each activity is labeled with a tuple—denoted as (i,h,k,t,f)—positioned directly on the activity within the charts. This tuple not only uniquely identifies the activity but also simplifies the task of tracking each operation through its sequential order and corresponding features.

Table 8. Production volume with no energy efficiency consideration

i	t	F	
		M	R
i=1	t=1	5	10
i=1	t=2	2	14
i=2	t=1	4	8
i=2	t=2	8	10

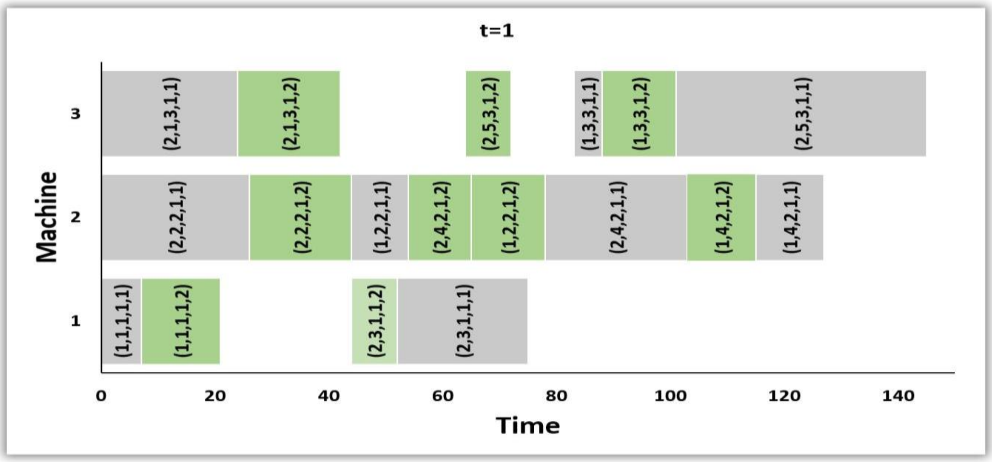

Figure 3. Scheduling with energy efficiency (t=1)

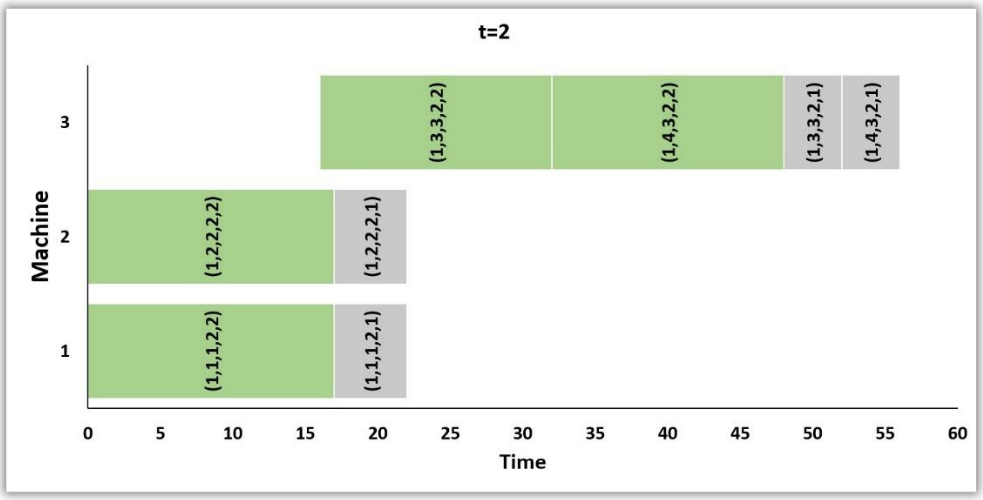

Figure 4. Scheduling with energy efficiency (t=2)

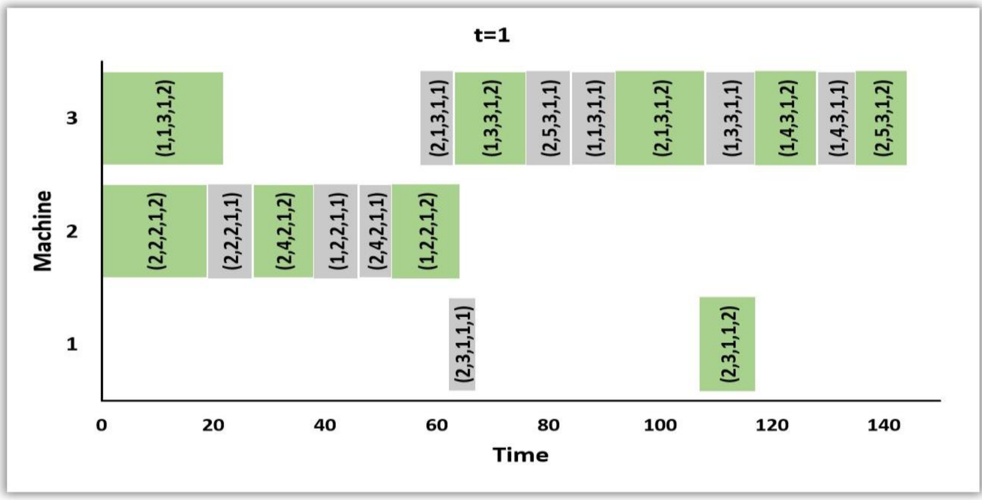

Figure 5. Scheduling activities in t=1

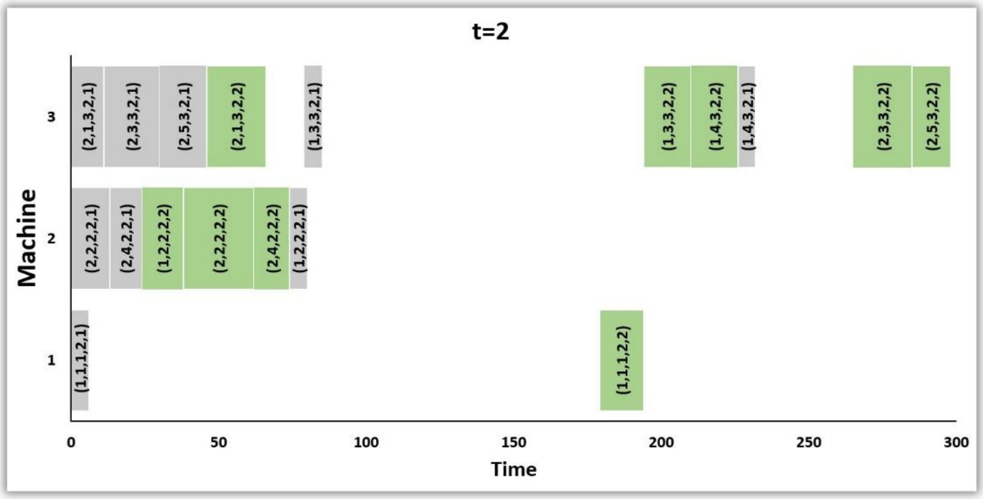

Figure 6. Scheduling of activities in t=2

Table 9. Comparison based on energy efficiency

	CPHS+CE	CE	CPHS	Time of idleness
With energy efficiency	1424	1018	406	92
Without energy efficiency	2338	2029	309	938

The table 10 provides the results of different optimization problems related to the scheduling of manufacturing and remanufacturing operations. The problems are distinguished by the values of the input parameters, such as the number of machines (K), time horizon (T), and number of activities of product types 1 and

2 (H1 and H2). The objective of each problem is to minimize the total cost.

The table shows the solution time, relative gap, and objective value obtained by solving each problem using the GAMS software. The relative gap represents the percentage difference between the obtained objective value and the best-known value for each problem.

By analyzing the table, we can observe that increasing the values of K, T, and H generally leads to an increase in the solution time and relative gap. This can be explained by the fact that the problem complexity increases with these parameters, which makes it harder to find the optimal solution.

In particular, problems 15, 16, and 17 having 4 machines, 4timeintervals, and varying numbers of activities have high relative gaps of 0.675, 0.57, and 0.58, respectively, indicating that the obtained solutions are far from the optimal ones. On the other hand, problems 1, 3, and 5 with lower values of K, T, and H have lower relative gaps of around 0.05, indicating that the solutions are relatively closer to the optimal ones.

Overall, the table provides insights into the trade-off between the problem complexity and the quality of the obtained solution, as well as the impact of the input parameters on the optimization results. Hence, this motivated the authors to develop meta-heuristic algorithms to cope with this problem.

Table 10. The results of solving problems in different sizes

Problem	|K|	|T|	|H|		Solution Time (GAMS)	Relative Gap (GAMS)	Objective Value (GAMS)
			H1	H2			
1	2	1	3	4	5	0.0476	492
2	2	2	3	4	13	0.05	1173
3	2	1	4	5	16	0.0476	579
4	2	2	4	4	20	0.046	1256
5	2	3	3	3	20	0.047	1454
6	2	2	4	3	24	0.049	1138
7	3	2	4	3	70	0.048	1105
8	3	1	4	5	100	0.1173	581
9	3	2	3	4	100	0.047	1133
10	3	2	4	5	100	0.277	1424
11	3	2	5	5	100	0.49	2051
12	2	3	4	4	100	0.37	2654
13	2	4	4	3	200	0.39	3133
14	3	2	4	4	200	0.35	1492
15	4	2	4	4	200	0.675	1461
16	4	3	4	4	200	0.57	1954
17	4	4	4	4	300	0.58	2382
18	4	2	5	5	300	0.15	2300
19	4	4	4	3	300	0.61	2616
20	3	5	3	4	300	0.5	3334
21	5	4	3	4	300	0.61	2893
22	4	3	3	3	400	0.27	1429
23	3	4	3	4	400	0.5	2890

4.2. Metaheuristic Solutions

In this research, we evaluated the effectiveness of three well-known metaheuristic optimization algorithms—Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Whale Optimization Algorithm (WOA)—in solving a specific and complex optimization problem. These algorithms were selected because they have been widely used in many optimization problems across different fields and have demonstrated strong and consistent performance. GA, PSO, and WOA were selected due to their proven effectiveness, diverse search strategies, and ease of hybridization with problem-specific heuristics. Although other algorithms could be used, the chosen ones offer a good balance of performance and implementation simplicity. Convergence was supported by multiple runs, heuristic enhancement, and comparison with deterministic solver results, ensuring robust and reliable solutions despite the stochastic nature of the methods.

Our primary aim was to assess and compare how well these algorithms perform in terms of both the quality of the solutions they generate and the amount of time they require to find these solutions.

To carry out this evaluation, we designed a set of test problems, each representing a realistic and challenging scenario. The data used for these tests are summarized in Table 11. This table provides detailed information for each test case, including the number of machines used, the time horizon considered for planning, and the number of tasks or activities that need to be scheduled for each product type. Additionally, the table reports the time each algorithm needed to find a solution, the relative gap between the algorithm's solution and the best-known solution (indicating how close the result is to the ideal), and the objective function values computed using GAMS software.

This section of the study includes a step-by-step description of how the experiments were structured. We explain how each algorithm was implemented, how the data was processed, and how the performance was measured. Following that, we present and discuss the results obtained from the experiments, allowing for a thorough comparison among the algorithms.

Importantly, each of the three algorithms was enhanced through a hybrid strategy. Rather than using the metaheuristic method alone, we combined it with a heuristic procedure. This hybrid approach was used both to create the initial set of possible solutions and to guide the generation of new solutions during the optimization process. This combination aims to improve both convergence speed and solution quality by leveraging the strengths of both heuristic and metaheuristic techniques.

The overall procedure followed by all three algorithms is illustrated in Figure 7. This figure provides a visual overview of the common structure used in the hybrid models, helping to clarify how the algorithms progress from initialization to solution evaluation and improvement.

1. Start by calculating the levels of on-hand inventory, incoming returned inventory, and any shortages. Use the predetermined production amounts (denoted as x) and apply Equations 5 and 6 to perform these calculations.
2. Assign the activities for manufactured products to different machines by making random selections for each time period.
3. Determine the duration each activity will take (represented as Pt) by consulting Equation 12.
4. Decide randomly on the order in which the activities will occur, taking into account both the activities and their calculated timings from the previous step.
5. Evaluate all of the solutions that have been generated. Arrange them in order from least to most based on the criteria specified by the objective function in Equation 1.
6. Examine whether the end conditions of the algorithm have been met, which entails either running out of the

allotted time or arriving at a solution more optimal than what was produced by the exact methods utilized in GAMS. If neither of these conditions is satisfied, the process moves onto step 7; else, the algorithm terminates.

7. Adapt the solution generation process for each specific algorithm. For the Genetic Algorithm, continue to generate the set number of new solutions by pairing two existing production sets (labeled $x^1 and\ x^2$) at random. This involves defining random parameters by determining values within chosen ranges, deduced through experimentation. Subsequently, create a matrix r whose size matches that of $x^1 and\ x^2$ and modify the existing solutions with the following (The random parameters $a = uniform(\alpha, \beta)$ and $b = uniform(\theta, \gamma)$ are first generated. The values α=-3, θ=1, β=1, and γ=2 are selected by trial and error in the present algorithm. Where $\rho = \text{round}(uniform(\text{a}, \text{a} + \text{b}))$.):

$$New\ solution\ 1: x_{new}^1 = \max(0, x^1 + r)$$
$$New\ solution\ 2: x_{new}^2 = \max(0, x^1 - r)$$

For PSO and WOA, modify solutions according to their standard protocols described in section 3.5.

8. Merge the new batch of solutions with the existing pool of solutions and return to step 2 for a new iteration.

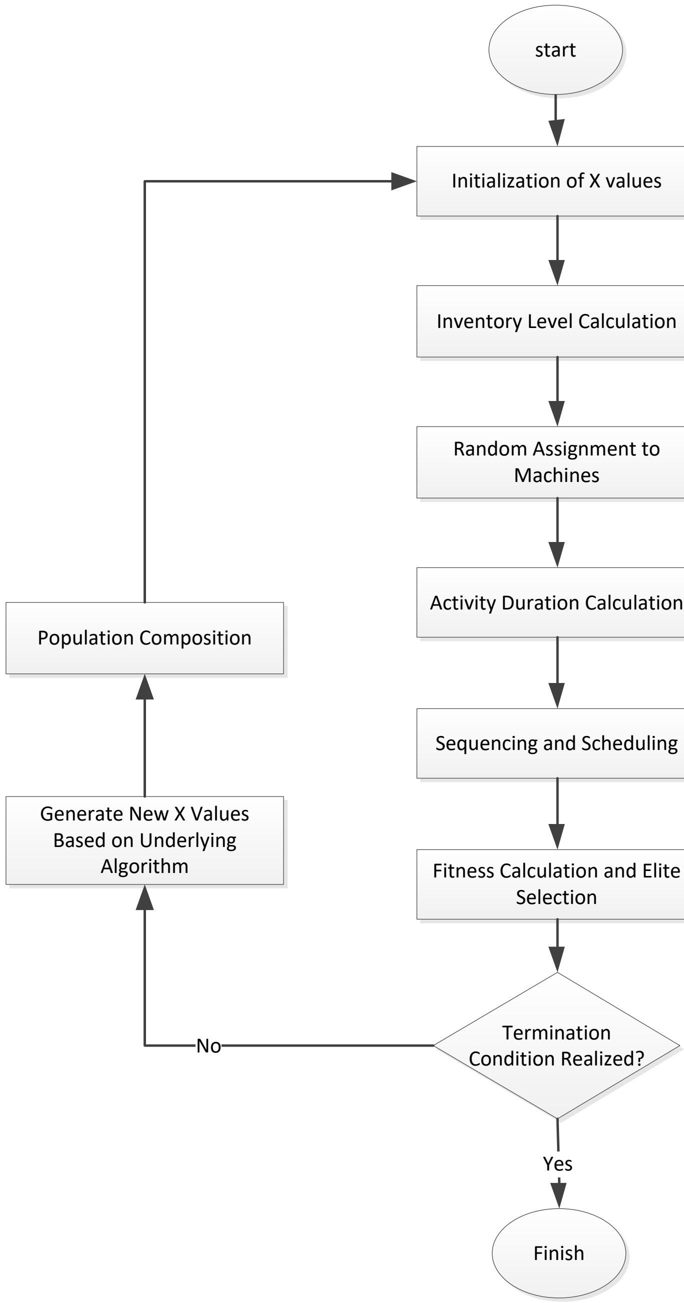

Figure 7. Flowchart of the proposed algorithms

In sum, this process is a structured exploration for solutions, fusing randomized initialization with informed updates based on algorithm-specific rules, and repeated cycles of evaluation and improvement until the algorithms either meet a predetermined time cap or yield an optimal solution.

4.3. Performance of metaheuristics

Table 11 presents the results of solving anSDFJP with energy efficiency consideration using three different meta-heuristics including GA, PSO, and WOA, and also GAMS software. The problem instances are labeled from 1 to 23, and for each instance, the table shows the number of machines (K), the time horizon (T), the number of operations per job (H), and the GAMS solution time.

For each problem instance, the table compares the solution quality and time for each algorithm against the GAMS solution. The relative gap column shows the percentage difference between the GAMS objective value and the optimal solution's objective value. The lower the relative gap, the closer the GAMS solution is to the optimal solution. In problems 1-7 and also for problem 9, the BARON solver in GAMS can find solutions with a relative gap of less than 5%, indicating that GAMS can find high-quality solutions close to the optimal solution. But for other problems, the problems cannot be solved with a relative gap of less than 5%.

The table also shows the objective value and time for each algorithm's solution. The termination condition for all the algorithms is set to reach the time that GAMS consumed to solve problems or achieve a solution better than that of GAMS. Note that the metaheuristic algorithms can achieve solutions better than GAMS because the relative gaps are not zero and thus GAMS solutions are not necessarily global optimum.

In terms of time, we can see that GA is generally faster than GAMS for most problems; i.e. it can achieve a solution as good as GAMS in less time. However, there are some exceptions where GAMS outperforms GA in terms of objective value, such as problems 4, 5, 16, 18, and 20. In addition, although in problems 1, 3, and 8 WOA has achieved the best solutions among the other three methods, however, it cannot obtain quality solutions compared to GAMS and GA in all other problems. Additionally, PSO has even weaker performance than WOA in most of the problems.

A comparison of all solution methods is presented visually in terms of objective value and solution time in Figures 8 and 9 and reveals that GA is (in most cases) the best solution approach. However, the choice between GAMS and GA depends on the specific problem and the trade-off between time and objective value.

Table 11. Performance of the proposed algorithms compared with GAMS

Problem ID	|K|	|T|	|H|		GAMS Time	Relative Gap	GAMS Objective Value	GA Objective Value	GA Time	WOA Objective Value	WOA Time	PSO Objective Value	PSO Time
			H1	H2									
1	2	1	3	4	5	0.0476	492	492	2	469.2	3	466.2	4
2	2	2	3	4	13	0.05	1173	1173	5	1384.4	8	1993.4	13
3	2	1	4	5	16	0.0476	579	579	9	573.6	12	739.6	16
4	2	2	4	4	20	0.046	1256	1292	20	1835.8	20	2976.8	20
5	2	3	3	3	20	0.047	1454	1728	20	2300	20	5189	20
6	2	2	4	3	24	0.049	1138	1138	11	1385.4	24	2761.4	24
7	3	2	4	3	70	0.048	1105	1105	7	1839.2	70	2666.2	70
8	3	1	4	5	100	0.1173	581	581	12	561.5	73	743.5	100
9	3	2	3	4	100	0.047	1133	1133	11	1217.2	100	2229.2	100
10	3	2	4	5	100	0.277	1424	1424	5	2780	100	2621	100
11	3	2	5	5	100	0.49	2051	2051	26	33014.4	100	19626.4	100
12	2	3	4	4	100	0.37	2654	2654	10	5251.2	100	6773.2	100
13	2	4	4	3	200	0.39	3133	3133	14	5777.8	200	9254.8	200
14	3	2	4	4	200	0.35	1492	1492	5	1783.6	200	2872.6	200
15	4	2	4	4	200	0.675	1461	1461	4	2189.4	200	2536.4	200
16	4	3	4	4	200	0.57	1954	2142	200	10314.6	200	5160.6	200
17	4	4	4	4	300	0.58	2382	2382	253	12577.8	300	9904.8	300
18	4	2	5	5	300	0.15	2300	2632	300	8663.2	300	28736.2	300
19	4	4	4	3	300	0.61	2616	2616	99	32187	300	52862	300
20	3	5	3	4	300	0.5	3334	4981	300	44849.6	300	58487.5	300
21	5	4	3	4	300	0.61	2893	2893	300	27869.9	300	33231.8	300
22	4	3	3	3	400	0.27	1429	1429	128	1673.5	400	3911.4	400
23	3	4	3	4	400	0.5	2890	2890	83	24085.4	400	33174.4	400

Figure 8 provides a comparative analysis of the optimal values yielded by metaheuristic approaches and GAMS software in solving optimization problems. The visual data indicates that, particularly with larger problem sets, the Genetic Algorithm frequently attains superior solutions in comparison to GAMS, and does so within a shorter time frame (bearing in mind that due to time constraints, the GAMS solutions may not be the best possible ones). Moreover, it is typically seen that the least favorable outcomes are associated with the Particle Swarm Optimization algorithm.

Moreover, Figure 9 compares the solution times of the metaheuristic algorithms with those obtained using GAMS software. This generally reveals that GA outperforms the others in terms of solution speed. Furthermore, the WOA and PSO have similar solution times with GAMS as they often fail to find solutions better or equivalent to those produced by GAMS in terms of the objective values.

Table 12 displays results from an Analysis of Variance (ANOVA) for objective values. The null hypothesis of this test is that all the solution methods (including GAMS and metaheuristic algorithms) have the same average of objective values. Although all the statistics of the ANOVA test are presented, we can make inferences just by using the P-value, which is the minimum significance level under which we can reject the null hypothesis. According to the P-value of 0.001 for this test, we can say that, under a 0.1 significance level, we can reject the null hypothesis of the ANOVA test (and equivalently, we can say that the solution methods have different performances concerning objective values). This can be seen graphically in Figure 10 which shows that GAMS and GA have lower total costs than WOA and PSO algorithms.

The result of the ANOVA test of the solution times is also presented in Table 13. The P-value of 0.058 for this test shows that, under a 0.1 significance level, we can again reject the null hypothesis of the ANOVA test (and equivalently, we can conclude that the solution methods have different performances concerningsolution times). This can be seen graphically in Figure 11 which shows that GA hasa lower average solution time than GAMS, WOA, and PSO algorithms.

Table 12. Analysis of variance for objective values

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Factor	3	2.09E+09	6.97E+08	6.06	0.001
Error	88	1.01E+10	1.15E+08		
Total	91	1.22E+10			

Table 13. Analysis of variance for solution times

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Factor	3	121631	40544	2.58	0.058
Error	88	1380905	15692		
Total	91	1502536			

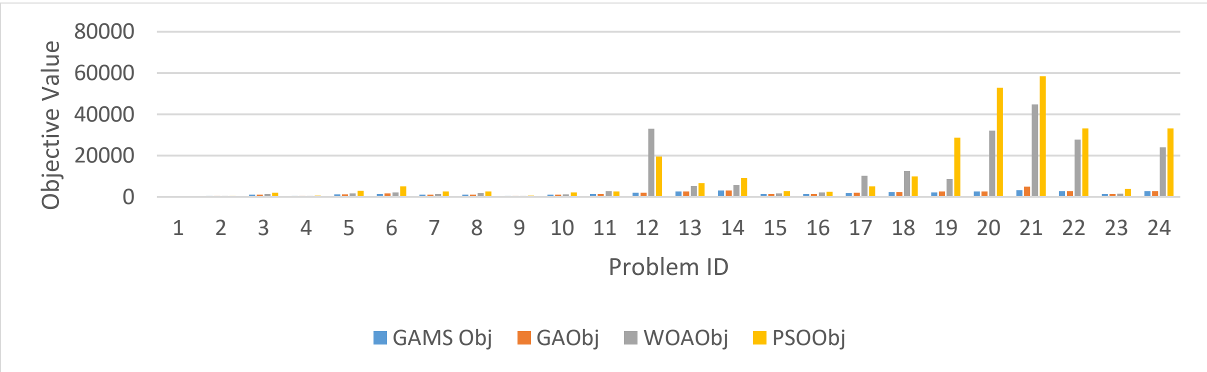

Figure 8. Comparison of the objective function obtained from the genetic algorithm and GAMS software

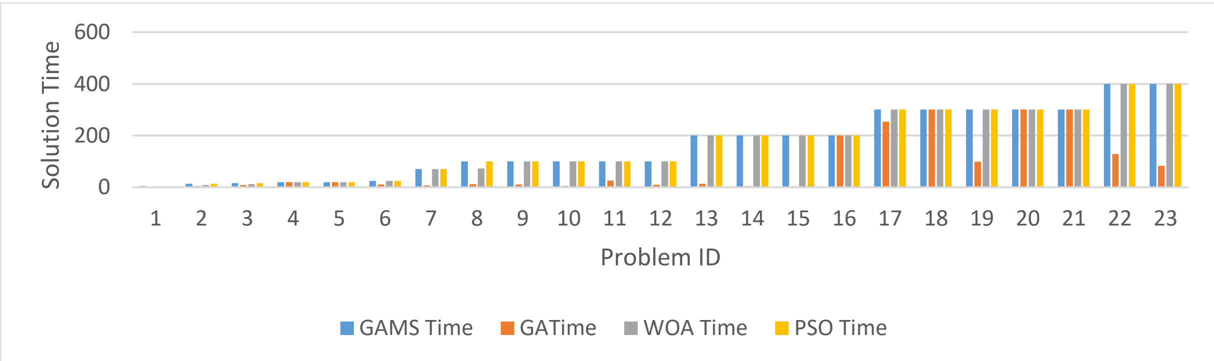

Figure 9. Percentage of reduction of CPU time using genetic algorithm

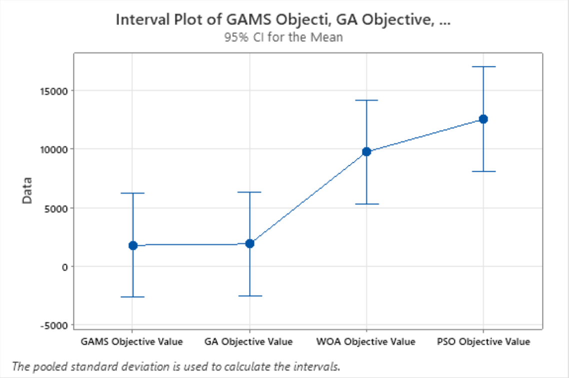

Figure 10. Average objective values

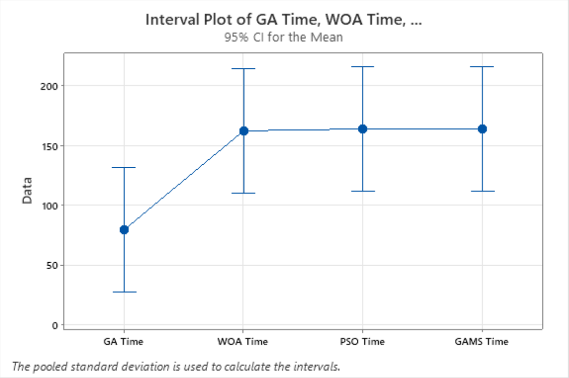

Figure 11. Average solution times

5. Conclusion

Recent research has shown a growing interest in SDFJSP. In this investigation, we constructed a mathematical framework aimed at both refining scheduling and lot-sizing challenges with a focus on energy efficiency. The model was framed as a MINLP and tackled through a ‘big bucket’ strategy, employing the GAMS software for solution finding.

To assess the effectiveness of this newly proposed model, we analyzed a case study excluding energy costs from the objective function. The findings made it clear that incorporating energy efficiency is impactful, resulting in considerable reductions in both the total cost and machine downtime. For dealing with larger-scale problems, we presented three evolutionary algorithm approaches: GA, WOA, and PSO. Notably, the genetic algorithms put to use differed from standard genetic approaches, utilizing alternative methods for generating new solutions in place of the usual cross-over and mutation techniques.

The results show that GAMS is not able to find high-quality solutions for some large problems. Among the metaheuristic algorithms, GAwas generally faster than GAMS and could achieve better solutions. However, there were some exceptions where GAMS outperformed the metaheuristic algorithms in terms of objective value. GA was found to be the best solution approach in most cases, but the choice between GAMS and GA depended on the specific problem and the trade-off between time and objective value.

This paper concentrated on tackling the complications arising from sequence dependence and energy efficiency within the SDFJSP. The results suggest that energy efficiency modeling can result in notable cost savings in this context. Specifically, accounting for energy costs in the objective function results in decreased energy costs but increased production costs, ultimately resulting in a reduction in total operational costs. These results highlight the importance of explicitly including sequence-dependent setup costs and energy costs SDFJSP.

The managerial insights derived from this research highlight the importance of incorporating energy costs into production planning decisions to achieve more cost-efficient and sustainable operations. The results show that traditional models focusing solely on production costs can significantly underestimate total expenses and lead to inefficient resource utilization, such as increased machine idle times. By adopting an integrated optimization approach that considers both manufacturing and energy consumption, managers can reduce overall costs, enhance machine utilization, and support environmentally conscious decision-making. These findings suggest that investment in energy-aware scheduling tools can provide long-term operational and financial benefits.

Finally, potential future research directions may include:

1. Extending the research to incorporate uncertainty in demand or production times, and evaluating the performance of metaheuristic algorithms under these conditions.
2. Investigating the applicability of the proposed solution approaches to other manufacturing systems, such as flow shops.
3. Comparisonof the different metaheuristic algorithms for solving the same problem instances and identifying which algorithms perform best under different conditions.
4. Evaluating the real-world effectiveness of the presented approach in manufacturing systems, and identifying any barriers to implementation in practice.

Reference

[1] Drexl, A. Kimms, “Lot sizing and scheduling—survey and extensions”, European Journal of Operational Research, Vol. 99, No. 2, 1997, pp. 221-235. https://doi.org/10.1016/S0377-2217(96)00318-5.

[2] Drexl, K. Haase, “Proportional lotsizing and scheduling,” International Journal of Production Economics, Vol. 40, No. 1, 1995, pp. 73-87.

[3] Jabbarzadeh, B. Fahimnia, F. Sabouhi, “Resilient and sustainable supply chain design: sustainability analysis under disruption risks,” International Journal of Production Research, Vol. 56, No. 17, 2018, pp. 5945-5968. https:// doi.org/ 10.1080/00207543.2017.1422464.

[4] Momenikorbekandi, M. Abbod, “A novel metaheuristic hybrid parthenogenetic algorithm for job shop scheduling problems: Applying optimization model,” IEEE Access, 2023. https://doi.org/10.1109/ACCESS.2023.3245678.

[5] Osati, E. Mehdizadeh, S. Ebrahimnejad, “Lot-sizing and scheduling in a flexible job-shop environment considering energy efficiency,” Journal of Industrial Engineering and Management Studies, Vol. 9, No. 2, 2022, pp. 129-147.

[6] Sifaleras, I. Konstantaras, N. Mladenović, “Variable neighborhood search for the economic lot sizing problem with product returns and recovery”, International Journal of Production Economics, Vol. 160, 2015, pp. 133-143. https://doi.org/10.1016/j.ijpe.2014.10.019.

[7] Van Zyl, O. Adetunji, "A lot sizing model for two items with imperfect manufacturing process, time varying demand and return rates, dependent demand and different quality grades", Journal of Remanufacturing, Vol. 12, No. 2, 2022, pp. 227-252. https://doi.org/10.1007/s13243-022-00260-z.

[8] Golany, J. Yang, G. Yu, "Economic lot-sizing with remanufacturing options," IIE Transactions, Vol. 33, No. 11, 2001, pp. 995-1004.

[9] Xin, S. Lu, Q. Wang, F. Deng, "A review of flexible job shop scheduling problems considering transportation vehicles," Frontiers of Information Technology & Electronic Engineering, Vol. 26, No. 3, 2025, pp. 332-353.

[10] N. Rosyidi, H. A. I. Permatasari, P. W. Laksono, "An integrated optimization model for procurement and production lot sizing and scheduling problems," Production Engineering Archives, Vol. 30, 2024.

[11] B. Fontes, S. M. Homayouni, J. C. Fernandes, "Energy-efficient job shop scheduling problem with transport resources considering speed adjustable resources", International Journal of Production Research, Vol. 62, No. 3, 2024, pp. 867-890. https://doi.org/10.1080/00207543.2023.2171685.

[12] D. Giglio, M. Paolucci, A. Roshani, "Integrated lot sizing and energy-efficient job shop scheduling problem in manufacturing/remanufacturing systems," Journal of Cleaner Production, Vol. 148, 2017, pp. 624-641. https://doi.org/10.1016/j.jclepro.2017.02.059.

[13] D. M. Carvalho, M. C. Nascimento, "Hybrid matheuristics to solve the integrated lot sizing and scheduling problem on parallel machines with sequence-dependent and non-triangular setup", European Journal of Operational Research, Vol. 296, No. 1, 2022, pp. 158–173. https://doi.org/10.1016/j.ejor.2021.03.024.

[14] Ahmadi, M. Zandieh, M. Farrokh, S. M. Emami, "A multi objective optimization approach for flexible job shop scheduling problem under random machine breakdown by evolutionary algorithms", Computers & Operations Research, Vol. 73, 2016, pp. 56–66.

[15] E. Mehdizadeh, A. Fatehi-Kivi, "A vibration damping optimization algorithm for solving the single-item capacitated lot-sizing problem with fuzzy parameters," International Journal of Industrial Engineering & Production Research, Vol. 28, No. 1, 2017, pp. 33-45. https://doi.org/10.22116/ijepr.2017. 92010.

[16] Pezzella, G. Morganti, G. Ciaschetti, "A genetic algorithm for the flexible job-shop scheduling problem," Computers & Operations Research, Vol. 35, No. 10, 2008, pp. 3202-3212.

[17] F. Soleimaninia, E. Mehdizadeh, "Mathematical Modeling and Solving Flexible Job-Shop Production Scheduling with Reverse Flows", Advances in Industrial Engineering, Vol. 52, No. 1, 2018, pp. 87-96. https://doi.org/10.1016/j.aien.2018.03.004.

[18] F. T. Chan, K. L. Choy, "A genetic algorithm-based scheduler for multiproduct parallel machine sheet metal job shop," Expert Systems with Applications, Vol. 38, No. 7, 2011, pp. 8703-8715. https://doi.org/10.1016/j.eswa.2011.02.043.

[19] F. T. S. Chan, T. C. Wong, L. Y. Chan, "Flexible job-shop scheduling problem under resource constraints", International Journal of Production Research, Vol. 44, No. 11, 2006, pp. 2071–2089. https://doi.org/10.1080/00207540500442365.

[20] F. Zhang, Y. Mei, S. Nguyen, M. Zhang, "Evolving scheduling heuristics via genetic programming with feature selection in dynamic flexible job-shop scheduling," IEEE Transactions on Cybernetics, Vol. 51, No. 4, 2020, pp. 1797-1811.

[21] B. T. Adams, "Energy-aware production planning with renewable energy generation considering combined battery: And hydrogen-based energy storage systems". Junior Management Science (JUMS), Vol. 10, No.1, 2025, pp. 267-291.

[22] G. Zhang, L. Gao, Y. Shi, "An effective genetic algorithm for the flexible job-shop scheduling problem," Expert Systems with Applications, Vol. 38, No. 4, 2011, pp. 3563-3573.

[23] C. Choi, D. S. Choi, "A local search algorithm for jobshop scheduling problems with alternative operations and sequence-dependent setups", Computers & Industrial Engineering, Vol. 42, No. 1, 2002, pp. 43-58. https://doi.org/10.1016/S0360-8352(01)00181-7.

[24] Kacem, S. Hammadi, P. Borne, "Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), Vol. 32, No. 1, 2002, pp. 1-13. https://doi.org/10.1109/TSMCC.2002.1005436.

[25] Mahdavi, B. Shirazi, M. Solimanpur, "Development of a simulation-based decision support system for controlling stochastic flexible job shop manufacturing systems," Simulation Modelling Practice and Theory, Vol. 18, No. 6, 2010, pp. 768-786. https://doi.org/10.1016/j.simpat.2010.03.010.

[26] C. Chen, K. H. Chen, J. J. Wu, C. W. Chen, "A study of the flexible job shop scheduling problem with parallel machines and reentrant process", The International Journal of Advanced Manufacturing Technology, Vol. 39, No. 3, 2008, pp. 344-354. https://doi.org/10.1007/s00170-007-1295-0.

[27] Gao, L. Sun, M. Gen, "A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems," Computers & Operations Research, Vol. 35, No. 9, 2008, pp. 2892-2907. https://doi.org/10.1016/j.cor.2007.04.016.

[28] H. Holland, "Genetic algorithms," Scientific American, Vol. 267, No. 1, 1992, pp. 66-73. https://doi.org/10.1038/scientificamerican0792-66.

[29] J. Hurink, B. Jurisch, M. Thole, "Tabu search for the job-shop scheduling problem with multi-purpose machines", Operations-Research-Spektrum, Vol. 15, No. 4, 1994, pp. 205-215. https://doi.org/10.1007/BF01928971.

[30] J. Kumar, G. Yadav, B. P. Agrawal, "A Hybrid Stochastic Optimization Model for Lot Sizing and Scheduling Problem," Cuestiones de Fisioterapia, Vol. 54, No. 2, 2025, pp. 2007-2018. https://doi.org/10.47393/cf.2025.06.

[31] J. Xie, L. Gao, K. Peng, X. Li, H. Li, "Review on flexible job shop scheduling," IET Collaborative Intelligent Manufacturing, Vol. 1, No. 3, 2019, pp. 67-77.

[32] Richter, M. Sombrutzki, "Remanufacturing planning for the reverse Wagner/Whitin models", European Journal of Operational Research, Vol. 121, No. 2, 2000, pp. 304-315. https://doi.org/10.1016/S0377-2217(99)00272-9.

[33] Z. Gao, P. N. Suganthan, M. F. Tasgetiren, Q. K. Pan, Q. Q. Sun, "Effective ensembles of heuristics for scheduling flexible job shop problem with new job insertion", Computers & Industrial Engineering, Vol. 90, 2015, pp. 107-117. https://doi.org/10.1016/j.cie.2015.08.018.

[34] M. Thi, T. T. Mai Anh, N. Van Hop, "An improved hybrid metaheuristics and rule-based approach for flexible job-shop scheduling subject to machine breakdowns", Engineering Optimization, 2022, pp. 1-21. https://doi.org/10.1080/0305215X.2022. 2037945.

[35] N. Xing, Y. W. Chen, P. Wang, Q. S. Zhao, J. Xiong, "A knowledge-based ant colony optimization for flexible job shop scheduling problems," Applied Soft Computing, Vol. 10, No. 3, 2010, pp. 888-896.

[36] L. Rocha, "A Simulation Based Solution Approach for the Capacitated Lot-Sizing Problem With Remanufacturing", 2023.

[37] L. Shen, S. Dauzère-Pérès, S. Maecker, "Energy cost efficient scheduling in flexible job-shop manufacturing systems", European Journal of Operational Research, Vol. 310, No. 3,

2023, pp. 992-1016. https://doi.org/10.1016/j.ejor.2023.01.010.

[38] L. Vidal, P. Piñeyro, O. Viera, "The integrated lot-sizing problem of production, remanufacturing and transportation, considering recovery targets", Journal of Remanufacturing, 2025, pp. 1-23. https://doi.org/10.1007/s13243-025-00270-8.

[39] F. Baki, B. A. Chaouch, W. Abdul-Kader, "A heuristic solution procedure for the dynamic lot sizing problem with remanufacturing and product recovery", Computers & Operations Research, Vol. 43, 2014, pp. 225–236. https://doi.org/10.1016/j.cor.2013.09.015.

[40] Hajibabaei, J. Behnamian, "Flexible job-shop scheduling problem with unrelated parallel machines and resources-dependent processing times: a tabu search algorithm", International Journal of Management Science and Engineering Management, Vol. 16, No. 4, 2021, pp. 242-253. https://doi.org/10.1080/17509653.2021.1894377.

[41] M. Rahimi, H. Fazlollahtabar, "Optimization of a Closed Loop Green Supply Chain using Particle Swarm and Genetic Algorithms", *Jordan Journal of Mechanical and Industrial Engineering*, Vol. 12, No. 2, 2018, pp. 77-91.

[42] M. Rohaninejad, A. Kheirkhah, P. Fattahi, "Simultaneous lot-sizing and scheduling in flexible job shop problems", The International Journal of Advanced Manufacturing Technology, Vol. 78, No. 1, 2015, pp. 1-18. https://doi.org/10.1007/s00170-014-6833-0.

[43] M. Rohaninejad, B. Vahedi-Nouri, R. Tavakkoli-Moghaddam, Z. Hanzálek, "A matheuristic approach for an integrated lot-sizing and scheduling problem with a period-based learning effect," Expert Systems with Applications, Vol. 269, 2025, 126234.

[44] M. Rohaninejad, B. Vahedi-Nouri, Z. Hanzálek, R. Tavakkoli-Moghaddam, "An integrated lot-sizing and scheduling problem in a reconfigurable manufacturing system under workforce constraints," International Journal of Production Research, Vol. 62, No. 11, 2024, pp. 3994-4013.

[45] M. Rohaninejad, R. Sahraeian, B. V. Nouri, "Multi-objective optimization of integrated lot-sizing and scheduling problem in flexible job shops", RAIRO-Operations Research, Vol. 50, No. 3, 2016, pp. 587-609. https://doi.org/10.1051/ro/2015025.

[46] Mat Ropi, H. Hishamuddin, D. A. Wahab, W. A. Jauhari, F. A. A. Rashid, N. K. Khamis, M. R. Abu Mansor, "Optimal lot-sizing decisions for a remanufacturing production system under spare parts supply disruption," Mathematics, Vol. 11, No. 19, 2023, pp. 4053. https://doi.org/10.3390/math11194053.

[47] Wang, Z. He, J. Sun, H. Xie, W. Shi, "A single-item uncapacitated lot-sizing problem with remanufacturing and outsourcing", Procedia Engineering, Vol. 15, 2011, pp. 5170-5178. https://doi.org/10.1016/j.proeng.2011.08.986.

[48] Brandimarte, "Routing and scheduling in a flexible job shop by tabu search", *Annals of Operations Research*, Vol. 41, No. 3, 1993, pp. 157–183. https://doi.org/10.1007/BF02098276.

[49] Brucker, R. Schlie, "Job-shop scheduling with multi-purpose machines", *Computing*, Vol. 45, No. 4, 1990, pp. 369–375. https://doi.org/10.1007/BF02241278.

[50] Piñeyro, O. Viera, "Inventory policies for the economic lot-sizing problem with remanufacturing and final disposal options," Journal of Industrial & Management Optimization, Vol. 5, No. 2, 2009, pp. 217.

[51] Piñeyro, O. Viera, "The economic lot-sizing problem with remanufacturing and one-way substitution," International Journal of Production Economics, Vol. 124, No. 2, 2010, pp. 482-488.

[52] H. Teunter, Z. P. Bayindir, W. V. Den Heuvel, "Dynamic lot sizing with product returns and remanufacturing", International Journal of Production Research, Vol. 44, No. 20, 2006, pp. 4377-4400. https://doi.org/10.1080/00207540600792065.

[53] Jans, Z. Degraeve, "Modeling industrial lot sizing problems: a review," International Journal of Production Research, Vol. 46, No. 6, 2008, pp. 1619-1643. https://doi.org/10.1080/00207540600852010.

[54] Sahraeian, M. Rohaninejad, M. Fadavi, "A new model for integrated lot sizing and scheduling in flexible job shop problem," Journal of Industrial and Systems Engineering, Vol. 10, No. 3, 2017, pp. 72-91.

[55] Teunter, O. Tang, K. Kaparis, "Heuristics for the economic lot scheduling problem with returns", International Journal of Production Economics, Vol. 118, No. 1, 2009, pp. 323-330. https://doi.org/10.1016/j.ijpe.2008.12.004.

[56] R. Zarrouk, I. E. Bennour, A. Jemai, "A two-level particle swarm optimization algorithm for the flexible job shop scheduling problem," Swarm Intelligence, Vol. 13, No. 2, 2019, pp. 145-168.

[57] Katoch, S. S. Chauhan, V. Kumar, "A review on genetic algorithm: past, present, and future," Multimedia Tools and Applications, Vol. 80, 2021, pp. 8091-8126. https://doi.org/10.1007/s11042-021-10685-x.

[58] Kumar, A. Goyal, A. Singhal, "Manufacturing flexibility and its effect on system performance", Jordan Journal of Mechanical and Industrial Engineering, Vol. 11, No. 2, 2017, pp. 105-112.

[59] S. Mahmud, R. K. Chakrabortty, A. Abbasi, M. J. Ryan, "Swarm intelligent based metaheuristics for a bi-objective flexible job shop integrated supply chain scheduling problems," Applied Soft Computing, Vol. 121, 2022, Article 108794. https://doi.org/10.1016/j.asoc.2022.108794.

[60] S. Mirjalili, A. Lewis, "The whale optimization algorithm," Advances in Engineering Software, Vol. 95, 2016, pp. 51-67. https://doi.org/10.1016/j.advengsoft.2016.01.008.

[61] S.Aghighi, S. T. A, Niaki, E. Mehdizadeh, & A. A.Najafi, "Open-shop production scheduling with reverse flows". Computers & Industrial Engineering, Vol. 153, 2021, pp. 107-157.

[62] M. Al-Turki, S. Arifusalam, M. El-Seliaman, M. Khan, "Resource allocation, batching and dispatching in a stochastic flexible job shop", Advanced Materials Research, Vol. 264, 2011, pp.1758–1763. https://doi.org/10.4028/www.scientific.net/AMR.264-265.1758.

[63] Xiong, D. Fu, "A new immune multi-agent system for the flexible job shop scheduling problem," Journal of Intelligent Manufacturing, Vol. 29, No. 4, 2018, pp. 857-873.

[64] Jiang, Z. Tian, W. Liu, Y. Suo, K. Chen, X. Xu, Z. Li, "Energy-efficient scheduling of flexible job shops with complex processes: A case study for the aerospace industry complex components in China," Journal of Industrial Information Integration, Vol. 27, 2022, Article 100293. https://doi.org/10.1016/j.jii.2022.100293.

[65] Liu, B. Liu, H. Yang, K. Luo, "Optimal production and maintenance strategies for manufacturing/remanufacturing leasing system considering uncertain quality and carbon emission," International Journal of Production Economics, Vol. 280, 2025, Article 109489. https://doi.org/10.1016/j.ijpe.2025.109489.

[66] Xu, D. Wang, M. Zhang, M. Yang, C. Liang, "Quantum particle swarm optimization with chaotic encoding schemes for flexible job-shop scheduling problem," Swarm and Evolutionary Computation, Vol. 93, 2025, p. 101836.