Journal of Manufacturing Systems 72 (2024) 287–307

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys

Technical paper

Data-driven simulation-based decision support system for resource allocation in industry 4.0 and smart manufacturing

Ehsan Mahmoodi [a], Masood Fathi [a,b,*], Madjid Tavana [c,d], Morteza Ghobakhloo [b], Amos H.C. Ng [a,b]

[a] Division of Intelligent Production Systems, School of Engineering Science, University of Skövde, 54128 Skövde, Sweden
[b] Division of Industrial Engineering and Management, Uppsala University, PO Box 534, Uppsala 75121, Sweden
[c] Business Systems and Analytics Department, Distinguished Chair of Business Analytics, La Salle University, Philadelphia, USA
[d] Business Information Systems Department, Faculty of Business Administration and Economics, University of Paderborn, Paderborn, Germany

ARTICLE INFO

Keywords:
Resource allocation
High-mix low-volume
Multi-objective optimization
Data-driven simulation
Decision support system
Industry 4.0
Meta-learning

ABSTRACT

Data-driven simulation (DDS) is fundamental to analytical and decision-support technologies in Industry 4.0 and smart manufacturing. This study investigates the potential of DDS for resource allocation (RA) in high-mix, low-volume smart manufacturing systems with mixed automation levels. A DDS-based decision support system (DDS-DSS) is developed by incorporating two RA strategies: simulation-based bottleneck analysis (SB-BA) and simulation-based multi-objective optimization (SB-MOO). To enhance the performance of SB-MOO, a unique meta-learning mechanism featuring memory, dynamic orthogonal array, and learning rate is integrated into the NSGA-II, resulting in a modified version of the NSGA-II with meta-learning (i.e., NSGA-II-ML). The proposed DSS also benefits from a post-optimality analysis that leverages a clustering algorithm to derive actionable insights. A real-life marine engine manufacturing application study is presented to demonstrate the applicability and exhibit efficacy of the proposed DSS and NSGA-II-ML. To this aim, NSGA-II-ML was tested against the original NSGA-II and differential evolution (DE) algorithm across a set of test problems. The results revealed that NSGA-II-ML surpassed the other two in terms of the number of non-dominated solutions and hypervolume, particularly in medium and large-sized problems. Furthermore, NSGA-II-ML achieved a 24% improvement in the best throughput found in the real case problem, outperforming SB-BA, NSGA-II, and DE. The post-optimality analysis led to the extraction of valuable knowledge about the key, influencing decision variables on the throughput.

Introduction

The fourth industrial revolution (I4.0), characterized by integrating innovative technologies and cutting-edge design principles, has substantially changed the conventional industrial landscape [1,2]. The implications of I4.0 on the manufacturing sector are extensive and profound, most notably a substantial enhancement in productivity and flexibility [3,4]. Moreover, the advent of I4.0 has precipitated a paradigm shift towards advanced operations management and analytical strategies [5]. Among the array of evolving analytical methodologies, data-driven simulation (DDS) has emerged as an effective and flexible approach for decision analytics and support in complex manufacturing environments [6,7].

DDS leverages the abundant data generated within the manufacturing environment to create accurate and dynamic models of manufacturing processes [8,9]. DDS can be considered a digital representation of the physical world [10] and can be utilized to predict system behavior, thereby offering foresight into potential issues and enabling the implementation of preemptive measures [11]. Moreover, they provide a risk-free environment for assessing different scenarios, thus facilitating the examination of various strategies and approaches [12, 13]. Additionally, DDSs enhance decision-making processes by offering data-supported insights and recommendations [14].

A crucial aspect of manufacturing wherein DDS proves invaluable is resource allocation (RA), a process inherently complex due to the dynamic nature of manufacturing operations [15]. RA refers to the systematic process of distributing resources among various performing units to achieve one or more objectives [16]. Appropriate RA is crucial to align with changes in the production plan and ensure its successful execution [17]. RA decisions significantly impact the most critical

* Corresponding author at: Division of Intelligent Production Systems, School of Engineering Science, University of Skövde, 54128 Skövde, Sweden.
E-mail address: masood.fathi@his.se (M. Fathi).

https://doi.org/10.1016/j.jmsy.2023.11.019
Received 8 August 2023; Received in revised form 2 November 2023; Accepted 27 November 2023

performance measure of production, namely throughput [18,19], and finding the appropriate allocation of resources to maximize throughout results in a highly complex decision-making problem known as the RA problem.

This study primarily focuses on exploring two critical applications of DDS in the domain of RA within manufacturing systems: simulation-based bottleneck analysis (SB-BA) and simulation-based multi-objective optimization (SB-MOO). SB-BA is a process that involves identifying resources or processes that are the constraints limiting the overall performance of the system [20]. This analysis is vital in any manufacturing setup as it aids in identifying areas that require improvement or modification to enhance system efficiency and productivity [21]. Conversely, SB-MOO aims to determine the most effective allocation of resources to maximize system performance. DDS in both these applications can offer a detailed and dynamic representation of the manufacturing system. It can support the evaluation of various allocation strategies and facilitate the identification of optimal solutions, enabling a more informed decision-making process.

The RA problem addressed in this study arises in industries with highly customized and complex products, such as marine and automotive engine manufacturers. Such products are manufactured by a mix of automated and manual processes, with production often involving high customization to meet specific customer requirements. This results in various product specifications and relatively low production volumes, as each product is unique and not mass-produced. The key to enhancing throughput in these systems lies in the strategic allocation of mobile resources across different segments of the production line. The production plans in these industries are dynamic and subject to frequent changes to accommodate varying product specifications. Therefore, a flexible RA strategy that can adapt to the changing needs of the production line is required. According to their flexibility and ease of reallocation, this study focused on mobile resources, namely operators and automated forklifts (AFLs), which can be easily reassigned according to the changing production plan.

In this paper, a DDS-based Decision-Support system (DDS-DSS) is proposed for the effective allocation of mobile resources in response to changing production plans and constraints imposed by the availability of resources. The proposed DDS-DSS is empowered by two mechanisms, namely simulation-based bottleneck analysis (SB-BA) and simulation-based multi-objective optimization (SB-MOO). However, each of these mechanisms, SB-BA and SB-MOO, face their own unique challenges when applied to resource allocation (RA), particularly in high-mix low-volume (HMLV) production systems [21].

One of the key challenges lies in the scalability of SB-BA [22,23]. As the complexity of the manufacturing system increases, the ability of SB-BA to effectively identify bottlenecks and constraints becomes a significant concern. Furthermore, given the complexity and frequency of RA problems in HMLVs, the performance of MOO algorithms is another critical issue. The need for efficient and high-quality solutions in a timely manner is paramount, yet achieving this in a constantly changing environment is a daunting task. Among the potential improvement mechanisms, the application of meta-learning stands out [24]. Meta-learning, as a higher-level learning approach, learns from past optimization experiences and applies this knowledge to improve the performance of the algorithm [25]. This approach can potentially improve the efficiency and quality of solutions in large optimization problems [26]. However, the application of meta-learning techniques, particularly in the context of RA, remains largely unexplored, especially within the HMLV production environments. This gap in research underscores the need for further investigation into the potential benefits and applications of meta-learning in this context.

This study aims to develop a DDS-DSS that efficiently addresses short-term resource allocation in HMLV production environments, particularly in industries with highly customized and complex products. Thus, the following research questions have been formulated to guide this study: (1) How can a DDS-based Decision-Support system (DDS-DSS) be developed to efficiently address short-term RA in HMLV production environments, particularly in industries with highly customized and complex products? (2) In the context of optimizing RA in HMLV production environments, how do the SB-BA and SB-MOO compare in terms of scalability and performance?".

Moreover, as an integrated part of the proposed DDS-DSS, the application of a clustering algorithm, namely density-based spatial clustering of applications with noise (DBSCAN), is investigated for performing post-analysis and extracting knowledge on the results of multi-objective optimization (MOO). The DBSCAN clusters the decision space into several regions of similar solutions, amongst which the decision-maker can choose the most desired region.

Given that the output of this study is an artifact composed of three distinct components - a simulation model, a bottleneck analysis tool, and an optimization algorithm – the methodology of design science (DS) is employed. DS is a research methodology that involves creating and evaluating artifacts designed to meet specific goals or solve particular problems. It is particularly suited to this study as it allows for the development and assessment of each component individually and their integration into a cohesive whole. DS methodology is characterized by iterative refinement, where the artifacts are continually evaluated and improved based on their performance in meeting the set objectives. This iterative process ensures that the final artifact effectively achieves its intended purpose and is robust in its design [27].

The rest of this paper is structured as follows. Section 2 presents a literature review to find and summarize the relevant research. Section 3 describes the problem addressed in detail. The detailed structure of the proposed DDS-DSS is elaborated in Section 4. The implication of the proposed DDS-DSS in a real-life application study is expounded in Section 5. The results of SB-BA and SB-MOO, including the performance evaluation of NSGA-II-ML, are respectively explained in Section 6. Section 7 further explains the post-optimality analysis and managerial insights. Finally, Section 8 provides a conclusion and highlights the insights gained from the study.

Literature review

This section reviews the existing research focusing on the DDS approach for RA in manufacturing industries.

Pierce and Yurtsever [28] introduced a data-driven system in the manufacturing context. They detailed Motorola's graphical manufacturing monitoring system (i.e., GraMMS). This data-driven system facilitated real-time visual monitoring of manufacturing data, including work-in-process (WIP), throughput, and dispatch rankings. The system enhanced wafer fab throughput by 15% and reduced cycle time by 20%, resulting in significant time and resource savings. Koyuncu et al. [29] developed a real-time dynamic, multi-scale simulation model that adaptively adjusted simulation fidelity by incorporating dynamic data. This model, applied in semiconductor manufacturing supply chain operations, utilized four heuristic algorithms for efficient computational resource management and inter-operable communication. Segura Velandia et al. [30] proposed a database system for handling the big data generated in printed circuit board manufacturing to improve product and process life cycle management. The proposed data storage and analysis approach improved the performance of their proposed simulation-based decision support system (DSS). This system improved data-exchange practices in the electronics manufacturing industry and was compatible with various tools and standards. Hussaini and Lahrman [31] demonstrated the effectiveness of a data-driven approach in solving real-life manufacturing problems through a comprehensive simulation model of a manufacturing facility. They emphasized the importance of DDS in the medical device industry, particularly for RA, performance evaluation, and operations excellence. Rashid et al. [32] presented an RA framework for modular construction, using a discrete event simulation (DES) model and a genetic algorithm (GA) to optimize worker assignments. Despite the challenges of MOO and adaptive GA for

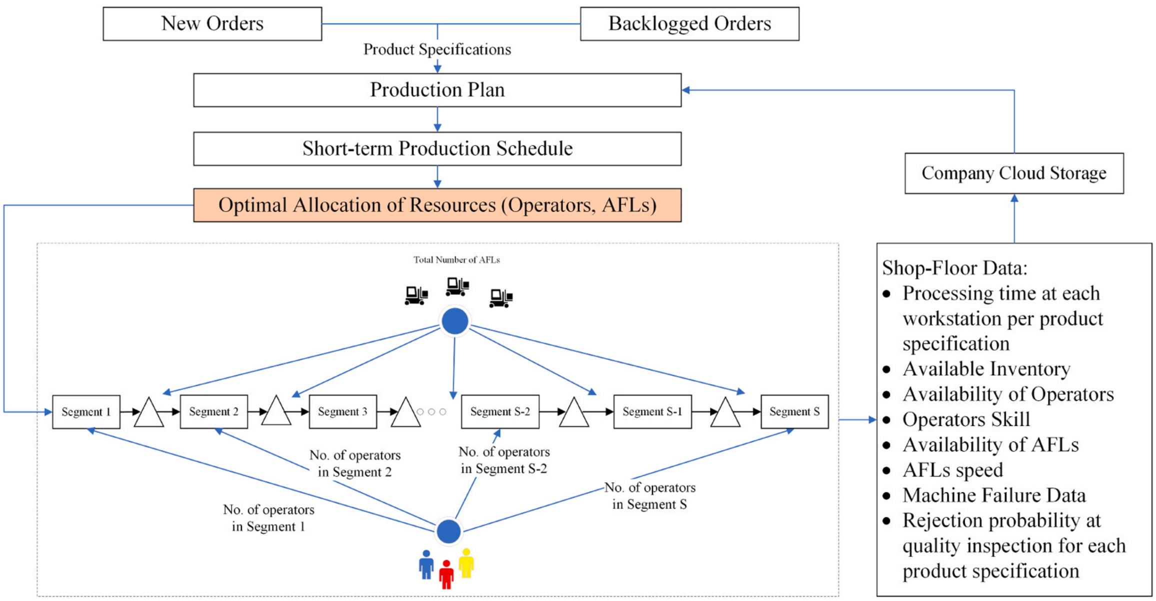


Fig. 1. The general configuration of an HMLV production line and production plan process.

real-time dynamic optimization, such a DDS approach resulted in a 15% make-span reduction.

Wang et al. [33] proposed a DDS framework for real-time process change, addressing throughput and resource utilization issues. The framework, integrated with Flexsim software, captured shop floor changes, although further research was suggested to develop plugin toolkits for manufacturing systems. Fani et al. [34] introduced a DDS for high-mix, low-volume (HMLV) environments, which required quick and reactive reconfiguration of production lines. The data-driven approach was implemented in a footwear industry SME, demonstrating its potential for improving performance without specific knowledge in building and validating simulation models. Luo et al. [35] highlighted the increasing trend of automated flexible production lines (AFPLs) in manufacturing enterprises. They proposed a data-driven cloud simulation architecture for AFPLs in smart factories, tested in a 3 C (computer, communication, consumer electronics) company's workshop. The architecture enabled real-time RA decisions, addressing a previously unmet need in the research on dynamic RA within AFPLs. More recently, Sakr et al. [36] provided a DDS for the fabrication areas in the semiconductor industry, characterized by high market demand and product mix. The authors proposed a DDS for dispatching and RA using reinforcement learning. The model they developed incorporated data-driven DES and agent-based simulation, with the agents employing Deep-Q-Network reinforcement learning. Their approach enhanced global system performance and RA, outperforming heuristics-based strategies. Consequently, this led to an improvement in the production performance.

The reviewed literature provided an overview of the evolution and application of DDS in various manufacturing contexts. From the early implementation of Motorola's GraMMS system, which significantly improved throughput and reduced cycle time, to the recent developments in AFPLs, DDS has been proven to be a powerful tool for enhancing manufacturing efficiency and performance. Such a literature review also shows the versatility of DDS, with successful applications in diverse industries such as semiconductor manufacturing, medical device production, modular construction industries, and footwear manufacturing. The studies above highlight the capacity of DDS to enable instantaneous monitoring, adaptable resource distribution, and effective decision-making in manufacturing operations.

Considering the notable progressions in DDS, there still exists a gap in research regarding the utilization of DDS in sectors that exhibit multi-phase production procedures and violent demand fluctuations. While Fani et al. [34] introduced a DDS in HMLV environments, their study was limited to a single SME in the footwear industry. Further research is needed to explore the applicability and effectiveness of DDS in HMLV industries, particularly those with more complex manufacturing processes. In addition, Luo et al. [35] and Sakr et al. [36] investigated the system throughput improvement via the dynamic distribution of tasks to available resources and the dynamic dispatching of products to specific workstations. Both studies assumed that production flow, including allocating tasks and dispatching products to fixed resources, can be dynamically altered. However, this assumption does not hold in all real-world production systems. In such cases, enhancing throughput necessitates the strategic reallocation of flexible and mobile resources rather than relying on fixed resources and changing the assignment of tasks to them. Thus, this study diverges from the prevailing literature by focusing on deploying mobile resources (i.e., operators and AFLs), leveraging their inherent flexibility to optimize production efficiency.

In light of the above explanations, the principal contribution of this study lies in developing a DDS-DSS to determine the optimal allocation of mobile resources across various segments of the production line while simultaneously accounting for fluctuations in demand. This research offers a unique perspective on improving manufacturing performance by emphasizing the strategic utilization of flexible resources through two distinctive approaches, namely SB-BA and SB-MOO. SB-BA identifies bottleneck areas and devises bottleneck-based strategies to assign resources to parts of the production line with a more urgent need. On the other hand, SB-MOO uses optimization to allocate resources near-optimally to different parts of the production line.

Problem description

In HMLV manufacturing environments, the RA problem involves the optimal assignment of mobile resources to various tasks or operations. This problem becomes particularly complex due to frequent changes in production plans, driven by varying product specifications and customer demands. The objective is to allocate these dynamic resources in a way that maximizes throughput while minimizing the total number of mobile resources allocated to the production line.

A schematic view of the problem addressed in this study is presented in Fig. 1. As depicted in this figure, the production plan is determined and updated based on new orders received, backlogged orders, and data

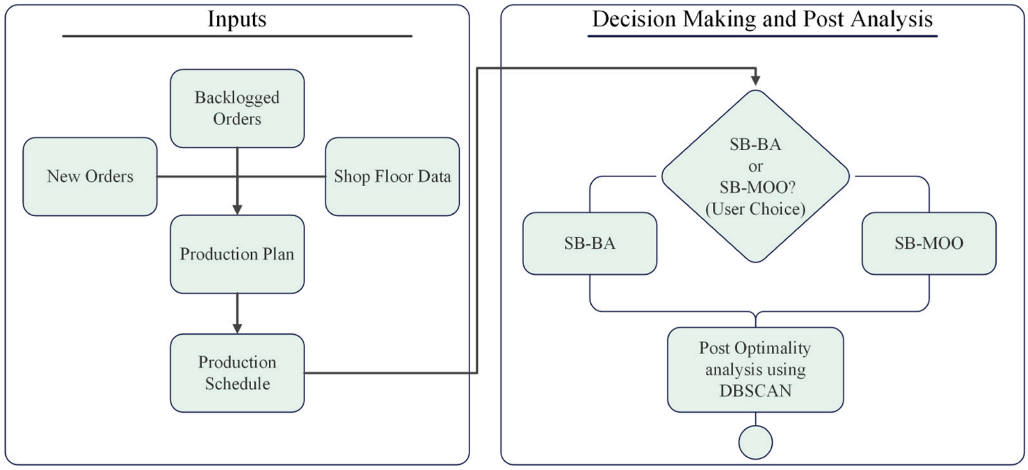


Fig. 2. The structure of the proposed DDS-DSS.

gathered from the shop floor. Afterward, the short-term production schedule is determined under the devised production plan. The next step is to find the optimal allocation of mobile resources to fulfill the short-term production schedule. This process repeats every time an update happens in coming orders. The problem addressed in this study occurs in HMLV production systems of complex and highly customized products, such as in the automotive or marine engine industry. In HMLV production systems, a high level of customization and frequent changes in product specifications make it difficult to maintain a smooth production flow. Consequently, the production planning and scheduling horizon is relatively short, making RA a frequent and challenging problem.

According to the complexity of production processes, such HMLV production systems benefit from various automation levels to maintain performance while gaining more flexibility. Using different levels of automation in different parts of the line makes it necessary to decouple the line at stations in which automation level changes, e.g., from automatic to manual or vice versa. These decoupling points are places to put buffers, which divide the line into several segments (S). A segment is then defined as a series of workstations connected without intermediate buffers. Each segment may contain one or more workstations. The number of operators required in each segment (Op_s) is dependent on the level of automation in that segment. Manual and collaborative segments are operated by full or partial involvement of operators, respectively. Fully automated segments do not need the involvement of operators. The processing time of each manual and semi-automated segment depends on the number of operators assigned to that segment. The problem addressed here is focused on the production lines in which AFLs accomplish material handling between decoupling points. The throughput of the line is highly dependent on the total number of available AFLs and the assignment of operators to line segments. With

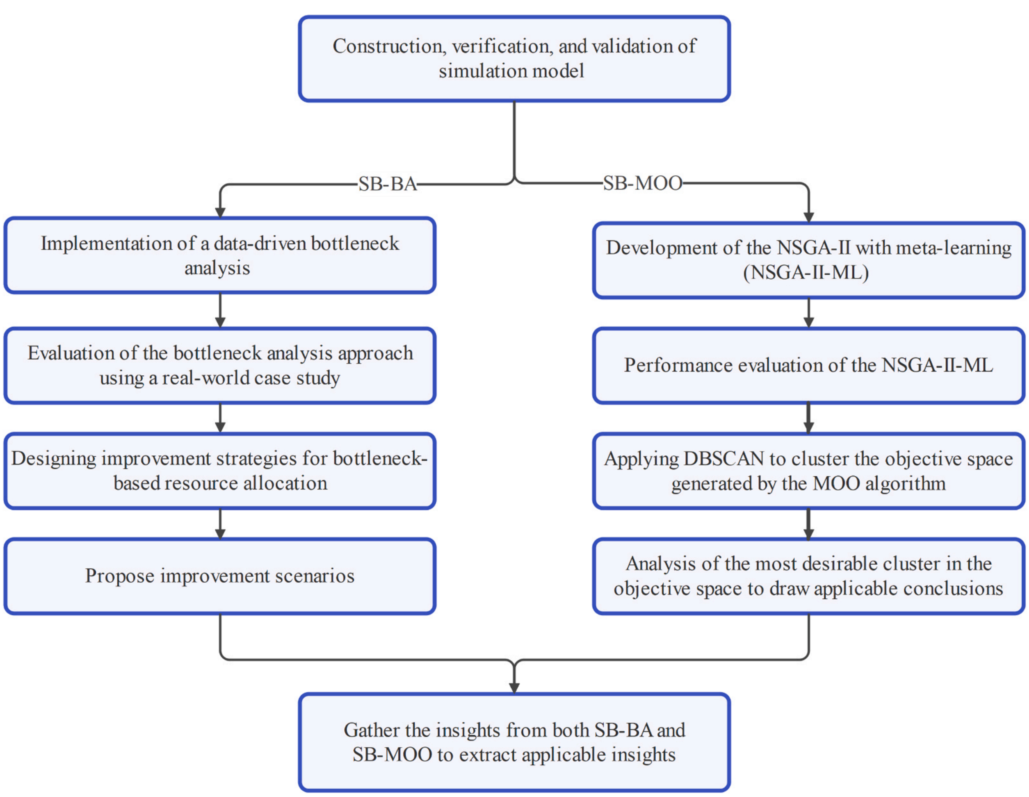


Fig. 3. The steps taken to realize and evaluate the proposed DDS-DSS.

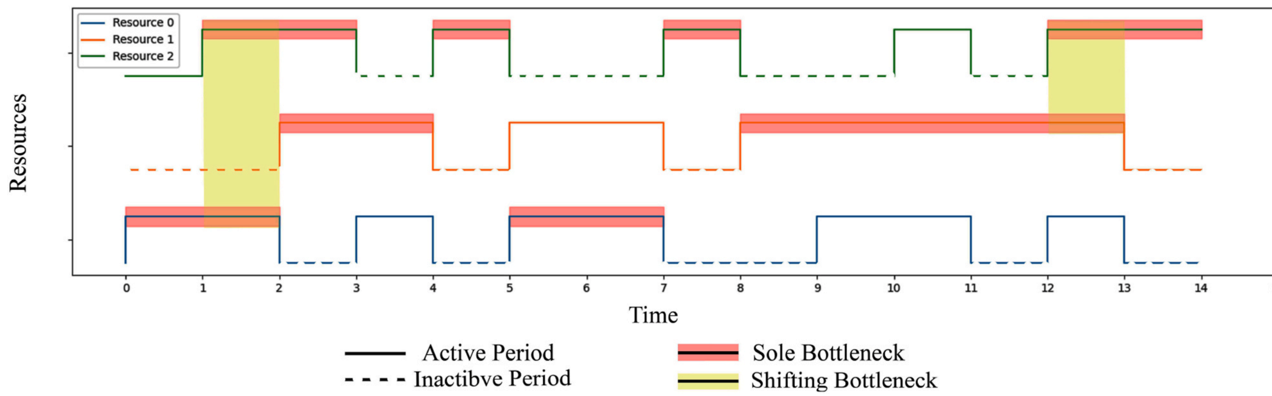


Fig. 4. A visualization of the active period method for resources using simulation outputs.

many AFLs, the throughput of the production line will be improved to a certain level by removing waiting time for material handling. However, the operating and maintenance costs will rise as well. Moreover, the risk of collision and the complexity of the work environment will increase with a higher number of AFLs. Hence, keeping the number of AFLs as low as possible is desired while ensuring it doesn't affect line throughput. Therefore, the objectives are maximizing throughput ($Max\,\mathbb{E}(Th)$), minimizing total number of operators ($Min \sum_{s\in S}(Op_s)$), and minimizing the number of AFLs ($Min\quad AFL$). The constraints of the problem include the lower and upper bound on the number of operators assigned to each segment ($Op_s^l \leq Op_s \leq Op_s^h$), the total number of available operators ($\sum_{s\in S}(Op_s) \leq$), and the maximum number of available AFLs.

Data-driven simulation-based decision support system

The proposed structure for the DDS-DSS is depicted in Fig. 2. To find the appropriate assignment of resources in the studied case, two different approaches, namely SB-BA and SB-MOO, are employed. FACTS Analyzer software [37] was used to build an efficient DES model for RA in the studied HMLV production system. The entering sequence, processing times at each workstation, the total number of available AFLs, time to failure, time to repair for each workstation, and travel time of AFLs are extracted from historical data and considered in the simulation model. The production plan, which includes the product variants and their specifications and the production steps for each variant, is given as inputs in the simulation period. The inputs can be updated according to a predefined update period or by the user command. Post-optimality analysis is performed by applying the DBSCAN algorithm to cluster the generated solutions according to the decision-maker's preferences to provide the managers with applicable rules. To realize the proposed DDS-DSS, the steps shown in Fig. 3 are undertaken.

Scenario analysis using SB-BA

Scenario analysis is creating and analyzing various possible outcomes of scenarios or situations using a model built based on real-world phenomena or artifacts. Through scenario analysis, it is possible to identify and evaluate potential risks and opportunities and develop strategies to respond to various possible outcomes. It also allows decision-makers to gain insights that inform their judgment and further optimize results. However, to make efficient scenarios, they must be built around the leverage points of a system. A leverage point in a system refers to a point at which a small change or intervention can lead to a significant impact or outcome. According to the theory of systems [38], the leverage point of a manufacturing system is its bottleneck. Thus, the scenarios in this study are designed to mitigate the bottlenecks of the systems. Hence, SB-BA is used to find the bottlenecks of the system. The simulation model imitates the behavior of the real system and generates the required data to perform BA.

Employing SB-BA for short-term resource allocation aligns with the existing literature on bottleneck management in complex manufacturing systems. Drawing parallels with the digital twin-based framework by Kumbhar et al. [39], the current study focuses on predicting future bottlenecks through simulation models, offering a proactive approach to resource reallocation. Furthermore, diagnostic insights generated by the data-driven approach proposed by Subramaniyan et al. [22] can be integrated into the simulation model to provide more comprehensive and effective resource allocation strategies. By incorporating diagnostic insights into the simulation model, it is possible to first identify the root cause of the bottleneck. This allows for determining whether resource allocation is the most suitable solution for the identified bottleneck. For instance, if the bottleneck is primarily due to maintenance issues, simply re-allocating resources may not address the underlying problem. In such cases, the simulation model can suggest alternative solutions, such as prioritizing certain maintenance activities.

The SB-BA used in this study benefits from the active period method (APM) [40] to identify and rank bottlenecks in the production system. APM divides the resource states into two categories, namely active and inactive. In APM, a resource is called inactive when it is starved (waiting for a part to arrive) or blocked (waiting for a finished part to leave that resource). The rest of the time is counted as active (Fig. 4). Using the APM, two numbers are calculated for each resource, the percentage of the sole bottleneck and the percentage of the shifting bottleneck. The bottleneck severity of a resource is then defined as the sum of sole and shifting bottleneck percentages for that resource. Following the identification of bottlenecks, several scenarios are devised for resource allocation to improve production throughput. In SB-BA, the general goal should be to prevent the most severe bottleneck (i.e., the resource with the highest value of bottleneck severity) from starving or being blocked. This can be done in the problem under study by relocating mobile resources in the production line.

Algorithm 1 describes the steps of the SB-BA approach to identify, rank, and mitigate bottlenecks using scenario analysis. In Algorithm 1, R is a set of resources involved in the production, T_{start} and T_{end} are the start and end of the duration to perform BA, t_o is the time duration between two consecutive observations, and N_o is the number of observations made over the time period $[T_{start}, T_{end}]$. At each observation, the state of the resources (zero if the resource is inactive, and 1 if the resource is active) is updated and stored in a matrix R_{states}, which is of the dimension $R \times N_m$. To perform SB-BA, active periods are extracted from the resource states matrix (R_{states}) using the "Extract active periods" function, which returns a list of "active periods" of each resource. Afterward, by applying the "Find bottlenecks" function to "Active periods," the resource with the longest active period is marked as a bottleneck at each t_o. Finally, the "Shifting bottleneck" function will get the “Bottleneck list” generated by the “Find bottlenecks” function and extract overlap time durations as shifting periods. Algorithm 1, equipped with a bottleneck visualization function, is implemented in Python (see [41]).

Algorithm 1. SB-BA resource allocation procedure.

```
Input: R = {r_1, r_2, ..., r_N }, T_start, T_end, t_o, N_o = (T_end − T_start)/t_o, R_states
function ExtractActivePeriods
    for each r in R_states do
        Active period (r) = uninterrupted active periods of each resource r in period [T_start, T_end]
    end for
    Active periods = [Active period(1), Active period(2), ..., Active period(R)]
    return Active periods
end function
function FindBottlenecks
    Get the list of Active periods
    Sorted active periods = Sort(Active periods) by starting time
    Bottleneck list = [], Last bottleneck end = None
    for each p and r in Sorted active periods do
        start = start time of (p), end = end time of (p)
        duration = end − start
        if Last bottleneck end = None then
            Bottleneck list = Bottleneck list + (p, r)
            Last bottleneck end = end
        else if (start > Last bottleneck end) OR (end > Last bottleneck end) then
            Bottleneck list = Bottleneck list + (p, r)
        end if
    end for
    return Bottleneck list
end function
function ShiftingBottleneck
    Bottleneck periods ← Bottleneck list
    Shifting periods = []
    for i in range (1, Bottleneck periods) do
        for j in range (i + 1, Bottleneck periods) do
            end^1 = end of i^th Bottleneck period
            start^2 = start of j^th Bottleneck period
            if end^1 > start^2 then
                overlap start = start^2
                overlap end = end^1
                Shifting periods = Shifting periods ← [(overlap start, overlap end)]
            end if
        end for
    end for
    return Shifting periods
end function
Improve the availability of resources in bottlenecks via scenario analysis
```

SB-MOO

The SB-MOO proposed in this study uses the simulation model outputs as inputs for the optimization algorithm. Upon receiving these outputs, the algorithm generates a revised assignment of resources, which the simulation model then assesses. This iterative process continues until the termination criterion of the optimization algorithm is met. In this paper, the outputs of the simulation model are the throughput of the production line, the number of operators assigned to each workstation, and the total number of AFLs.

NSGA-II-ML

This study uses an enchanted non-dominated sorting genetic algorithm (NSGA-II) as the optimization algorithm, incorporating a customized meta-learning mechanism to optimize its performance. NSGA-II is amongst the most commonly used evolutionary algorithms that can solve multi-objective problems [42].

The pseudo-code of the NSGA-II-ML is presented in Algorithm 2. More details about the NSGA-II-ML, including solution representation, non-dominated sorting, crowding distance, meta-learning with memory, dynamic orthogonal arrays, and learning rate, crossover, and mutation, are described, respectively. The notation used in Algorithm 2 is as follows: N is the population size, *gen* is the number of generations, α and p_c are crossover alpha and crossover probability, respectively, p_m is mutation probability, *learning_timeout* is the time given to the learning process in each iteration of NSGA-II-ML, *Th*, *Ops*, *AFLs* are objectives, and *Meta_learningParameters* are the parameters of the meta-learning mechanism and are described in more detail.

The NSGA-II algorithm is implemented by creating a first initial population of parents (P). Each solution in the population is assigned a Pareto front through the sorting algorithm. A binary tournament, crossover, and mutation are used in this step to create an offspring Q with a size of N. The algorithm then incorporates a meta-learning mechanism, which runs for a specified duration within each generation. This mechanism adjusts the crossover and mutation probabilities based on the learning from previous generations, thereby improving the algorithm's performance over time.

The hybrid population is sorted based on the crowding comparison operator, and then the N best solutions are taken as the future population (P_{t+1}). By using the crowding comparison operator and selecting a binary tournament, this algorithm guarantees the population variety of each generation. Non-dominated sorting will result in multiple Pareto fronts of the combined population ($R_t = P_t\, UQ_t$). The best solutions are grouped in the first front, namely F1, followed by the second, third, and other fronts generated. This selection method ensures that elite members of the latest generation are not removed, resulting in higher stability and convergence.

Algorithm 2. NSGA-II-ML.

Input: N, gen, α, p_c, p_m, Th, Ops, $AFLs$, $learning\_timeout$, $Meta\_LearningParameters$:{lr, Mem_{size}, σ_{high}, σ_{low}, $Orth_{large}$, $Orth_{medium}$, $Orth_{small}$, $learning\_iters$}
Initialize population P_0 of size N
Convert the values of each cell in each individual p to range $[0, 1]$
Calculate fitness values Th, AFL, and Op_s
for each x_i in P_0 **do**
    Non-dominated sorting (nds): $F_0 \leftarrow nds(P_0)$
end for
for each x_i in each front **do**
    Calculate crowding distance for F_0 (cd(F_0))
end for
$t = 0$
while $t < gen$ **do**
    while $learning\_duration < learning\_timeout$ **do**
        Perform Meta Learning (P_t, $Meta\_LearningParameters$)
    end while
    Perform binary tournament selection on P_t
    Apply crossover and mutation to create offspring Q_t
    for each pair of parents (p_1, p_2) in the mating pool **do**
        Perform BLX-α crossover: BLx-$\alpha(p_1, p_2)$
        Apply Gaussian mutation
    end for
    $R_t \leftarrow P_t \cup Q_t$
    $F_t \leftarrow NDS(R_t)$
    for each x_i in each front **do**
        perform cd(F_t)
    end for
    Construct the next population P_{t+1} by selecting the best N individuals from F_t
    $t = t + 1$
end while
return the final non-dominated set of solutions.

Solution representation

Meta-heuristic algorithms are most effective when the solution is represented effectively [43]. In this study, a solution representation for the defined problem is designed. This representation has a length of $N_S + 1$, in which N_S is the number of segments, and the last part is the number of AFLs used in the line. A sample of the solution representation is presented in Fig. 5.

Fast non-dominated sorting and crowding distance

Using Fast Non-Dominated Sorting for each solution p, two factors are determined; S_p, a set of solutions dominated by solution p, and n_p, the number of solutions that dominate solution p. For a population with the size N and a problem with m objective functions, the computational complexity of the non-dominated sorting is equal to $O(mN^2)$. First Pareto front, F_1, is composed of the solutions that their $n_p = 0$. In this step, the n_j for each member of S_p, belonging to F_1, is reduced by one, and members with $n_j = 0$ will form F_2. In the next step, the members of F_1 are left from the calculations, and the same procedure will happen on the F_2. Using non-dominated sorting, the solutions will be ranked on several fronts, with Rank 1 being the best, followed by Rank 2, etc. Algorithm 3 presents the pseudo-code of the non-dominated sorting algorithm.

Algorithm 3. Non-dominated sorting.

for each solution p in R **do**
  $S_p = []$, $n_p = 0$
  for each solution q in R **do**
    if p dominates q **then**
      Add q to S_p
    else if q dominates p **then**
      Increment n_p by 1
    end if
  end for
  if $n_p = 0$ **then**
    $F_1 \leftarrow p$
    $p's$ rank $= 1$
  end if
end for
Initialize $i = 1$
while F_i is not empty **do**
  Initialize $Q = []$
  for each solution p in F_i **do**
    for each solution q in S_p **do**
      Decrement n_q by 1
      if $n_q = 0$ **then**
        $q's$ rank $= i + 1$
        Add q to Q
      end if
    end for
  end for
  $i = i + 1$
  $F_{i+1} = Q$
end while
$\mathbb{F} = \{F_1, F_2, ..., F_i\}$

NSGA-II uses a measure called crowding distance to determine the density index of each solution. Crowding distance is used to adjust the variety of the solutions generated in each iteration of the optimization algorithm. The average distance of points surrounding a solution is calculated to find the density index for a solution. This value is used as an estimated value for the perimeter of the cuboid surrounding a particular solution without including other points [44]. The pseudo-code for crowding distance calculation is shown in Algorithm 4, in which F represents fronts stored in $\mathbb{F}$ (returned by Algorithm 3).

Algorithm 4. Crowding distance.

for each front F in $\mathbb{F}$ **do**
  for each solution p in F **do**
    $p's$ distance $= 0$
  end for
  for each Obj in objectives **do**
    Sort F according to objective o
    $F[1]'s$ distance $= F[N]'s$ distance $= \infty$
    for each solution p in F from 2 to $N - 1$ **do**
      $p's$ distance $= p's$ distance $+ \frac{F[i+1].Obj - F[i-1].Obj}{\max(Obj) - \min(Obj)}$
    end for
  end for
end for

Meta-learning with memory, dynamic orthogonal arrays, and learning rate

The meta-learning algorithm begins by taking some required inputs, which are the current population P_t, crossover probability p_c, mutation probability p_m, and Meta_learning paramteres. Meta_laerning parameters are learning rate ($lr = 0.85$), memory size ($Mem_{size} = 10$), thresholds for the standard deviation of population ($\sigma_{low} = 0.2, \sigma_{high} = 0.8$), the size of orthogonal array ($Orth_{large} = 10, Orth_{medium} = 5, Ort = 5$), and the number of learning iterations ($learning\_iters = 5$). The algorithm then calculates the standard deviation (σ) of the fitness values of the current population P_t. Depending on the value of this standard deviation, it selects an appropriate size (S) for the orthogonal array, which is a statistical method of experimental design providing a set of well-balanced (orthogonal) data. The pseudo-code of the Meta-learning algorithm with memory, dynamic orthogonal array, and learning rate is provided in Algorithm 5.

The algorithm then checks if there is any existing memory. If there is, it calculates the average crossover and mutation probabilities from the different combinations of crossover and mutation probabilities and observe their effects on the population's fitness.

Algorithm 5. Meta-learning with memory, dynamic orthogonal, and learning rate.

Input: P_t, p_c, p_m, lr, Mem_{size}, σ_{high}, σ_{low}, $Orth_{large}$, $Orth_{medium}$, $Orth_{small}$, $learning\_iters$
fitness_values $\leftarrow$ fitness values for each individual in P_t
$\sigma \leftarrow$ std(fitness_values)
if $\sigma > \sigma_{high}$ **then**
  S $\leftarrow Orth_{large}$
else if $\sigma < \sigma_{low}$ **then**
  S $\leftarrow Orth_{small}$
else
  S $\leftarrow Orth_{medium}$
end if
if $memory \neq \emptyset$ **then**
  $size_m \leftarrow \text{size}(memory)$
  $p_{c_{avg}} \leftarrow average(p_{c_i})$
  $p_{m_{avg}} \leftarrow average(p_{m_i})$
  $O \leftarrow$ **getOrtogonalArray**$(p_{c_{avg}}, p_{m_{avg}}, S)$
else
  $O \leftarrow$ **getOrtogonalArray**$(initp_c,\ initp_m,\ \text{S})$
end if
$std\_devs \leftarrow []$
for each $o \in O$ **do**
  for $i = 1$ to $learnng\_iters$ **do**
    $[Th, Ops, AFls] \leftarrow$ Append population from NSGA-II (P_t, p_c and p_m from o)
  end for
  $[std\_devs, o] \leftarrow$ Append $\max(\sigma_{Th}, \sigma_{Ops}, \sigma_{AFLs})$
end for
$\delta \leftarrow$ **get_best_param**$(std\_devs,\ o)$
Update p_c and p_m using learning rate (lr):
  $p_c \leftarrow lr \cdot \delta[0] + (1 - lr) \cdot p_c$
  $p_m \leftarrow lr \cdot \delta[1] + (1 - lr) \cdot p_m$
  $memory \leftarrow$ Append (p_c, p_m)
if $size_m > Mem_{size}$ **then**
  Remove the oldest element from $memory$
end if

memory and generates an orthogonal array based on these averages. If there is no existing memory, it generates an orthogonal array based on the initial crossover and mutation probabilities. This orthogonal array is used to explore the parameter space in a balanced way, ensuring that the algorithm doesn't focus too much on one area of the parameter space.

The next part of the algorithm involves a learning process. For each orthogonal array, the algorithm runs a certain number of learning iterations. In each iteration, it appends a new population from NSGA-II using the crossover and mutation probabilities from the orthogonal array. It then calculates the maximum standard deviation of the fitness values of the three populations (*Th*,*Ops*,*AFls*) and appends this to a list of standard deviations. This process allows the algorithm to explore

After the learning process, the algorithm uses the *Get best parameters* function to find the best parameters (Algorithm 6). This function normalizes the standard deviations to a range between − 1 and 1 and finds the index of the closest value to zero (*closest_index*). This index corresponds to the best orthogonal array, which contains the best parameters. The algorithm then updates the crossover and mutation probabilities using a learning rate (*lr*) and the best parameters. It also appends these new probabilities to the memory. If the memory size exceeds the specified limit, it removes the oldest element. This process allows the algorithm to gradually adjust the crossover and mutation probabilities based on the results of the learning process, improving its performance over time.

Algorithm 6. Get best parameters.

Require: $std\_devs, orthogonal$

$normalized\_std\_devs \leftarrow 2 * \left(\frac{std\_devs - \min(std\_devs)}{\max(std\_devs) - \min(std\_devs)}\right) - 1$

$closest\_index \leftarrow \underset{i}{\operatorname{argmin}} \left|\frac{normalized\_std\_devs[i]}{1}\right|$

$best\_orthogonal \leftarrow orthogonal[closest\_index]$

return $best\_orthogonal$

The *Get best parameters* function is a crucial part of the algorithm. It takes the standard deviations and the orthogonal array as inputs. The function first normalizes the standard deviations to a range between − 1 and 1. It then finds the index of the value in the normalized array that is closest to zero. This index is used to select the corresponding element from the orthogonal array, which is returned as the best parameters. This function essentially identifies the set of parameters that resulted in the most stable (lowest standard deviation) performance during the learning process.

Crossover and mutation operators

Crossover is an operator used in GAs and evolutionary computation to combine the genetic information of two parents to generate new offspring. This process can generate offspring exhibiting traits that fall beyond the range of values defined by the parental genes. This study employed a crossover suitable for GAs with real-valued parameters named Blend-Alpha (BLX-α) [45]. The details of the applied crossover are presented in Algorithm 7. $p1$, and $p2$ are two randomly selected parents, the parameter α governs the extent of extrapolation, and p_c is crossover probability. Although this approach can facilitate a more efficient search space exploration, it may generate offspring that conflict with the constraints of the problem, necessitating supplementary handling methods. In this study, the gene values are in the range [0,1]. Thus, the newly generated values are clipped to this range using the $\min(\max(c, 0), 1)$ formula. The value of α is set to be 0.5, as proposed by several studies [46].

Algorithm 7. BLX-α Crossover.

Input: $p1, p2, \alpha, p_c$
$r \leftarrow \mathrm{rand}(0, 1)$
for each cell i in $p1, p2$ **do**
  if $r > p_c$ **then**
    $c_{\min} \leftarrow \min(p1[i], p2[i]) - \alpha \cdot \mathrm{abs}(p1[i] - p2[i])$
    $c_{\max} \leftarrow \max(p1[i], p2[i]) + \alpha \cdot \mathrm{abs}(p1[i] - p2[i])$
    $c1[i] \leftarrow \min(\max(\mathrm{rand}(c_{\min}, c_{\max}), 0), 1)$
    $c2[i] \leftarrow \min(\max(\mathrm{rand}(c_{\min}, c_{\max}), 0), 1)$
  end if
end for

The mutation is a genetic operator utilized by evolutionary algorithms to preserve diversity and explore new regions of the search space. It includes introducing random modifications to the genetic information of a population member to develop new solutions. Mutation aims to add diversity to a population of solutions and prevent premature convergence. This study benefits from the Gaussian mutation, which involves adding a stochastic value selected from a Gaussian distribution to the pre-existing value of a gene [47]. The details of the applied mutation are described in Algorithm 8. The degree of alteration to a specific gene is determined by the standard deviation of the Gaussian probability distribution using this mutation. The utilization of the Gaussian mutation operator is frequently observed due to its ability to facilitate a substantial likelihood of minor alterations, owing to the Gaussian distribution's characteristics, while also enabling sporadic significant leaps.

Algorithm 8. Gaussian mutation.

Input: individual p, p_m
for each cell in p **do**
  $r \leftarrow \mathrm{rand}(0, 1)$
  if $r > p_m$ **then**
    $c \leftarrow$ cell value
    $s \leftarrow N(0, \sigma)$
    $c \leftarrow c + s$
    if c is outside the allowed range for the cell **then**
      $c \leftarrow \min(\max(c, 0), 1)$
    end if
    Update the cell value with c
  end if
end for

Post-optimality analysis

Post-optimality analysis is usually performed on the results of MOO to extract applicable knowledge from a portion of decision space that is more favorable to decision-makers. Similar to Bandaru et al. [48], this study applies a posterior approach, in which the DBSCAN clustering algorithm is used to cluster the decision space according to the decision-makers' preferences [49]. Then, frequency analysis is performed over the solutions, falling in the desirable cluster(s) to generate applicable rules for the values of decision variables.

DBSCAN is a clustering technique that does not adhere to the constraints of convexity and isotropy imposed by other methods like K-means. The adaptability of DBSCAN accommodates the intricate and diverse structures that frequently arise in MOO situations. In such scenarios, the Pareto-efficient solutions frequently exhibit non-linear and non-uniformly shaped clusters in the objective space, which can be effectively detected and clustered by the DBSCAN [50].

For better clarification, the pseudo-code of the DBSCAN algorithm is provided in Algorithm 9. The input parameters of the algorithm include dataset D (a dataset of values of objective functions generated by the MOO algorithm), a radius epsilon, and a minimum number of points MinPts. The algorithm commences with the iteration of every unvisited point in the dataset, subsequently designating it as visited. The algorithm identifies the epsilon-neighborhood of a given point P, which encompasses all points within a distance of eps from P. It subsequently verifies whether this neighborhood comprises a minimum of “MinPts” points. If it does, a novel cluster C is initiated and subsequently augmented by including all data points deemed density-reachable from P, denoting all the points that are part of the same cluster as P. The algorithm proceeds with the expansion of the cluster until no additional density-reachable points are discovered. Points that are not density-reachable from any other point are classified as noise. The function “RegionQuery” yields points located within the epsilon-neighborhood of point P. The function “ExpandCluster” is designed to incorporate a given point P into a designated cluster C and subsequently expand the cluster by including all points considered density reachable. The termination of the algorithm occurs upon the visiting of all points.

Algorithm 9. DBSCAN.

```
Input: D (dataset of objective values found by MOO), eps = 0.025, MinPts = 5, C = 0
for each unvisited point P in dataset D do
    Mark P as visited
    NeighborPts = RegionQuery(P, eps)
    if |NeighborPts| < MinPts then
        Mark P as Noise
    else
        C = C + 1
        ExpandCluster(P, NeighborPts, C, eps, MinPts)
    end if
end for
Input: P, NeighborPts, C, eps, MinPts
Add P to cluster C
for each point P' in NeighborPts do
    if P' is not visited then
        Mark P' as visited
        NeighborPts' = RegionQuery(P', eps)
        if |NeighborPts'| ≥ MinPts then
            NeighborPts = NeighborPts joined with NeighborPts'
        end if
    end if
    if P' is not yet a member of any cluster then
        Add P' to cluster C
    end if
end for
Input: P, eps
Return all points within P''s eps-neighborhood (including P)
```

A real-world application study

This study addresses a real-world RA problem in an HMLV marine engine manufacturer in Sweden. The factory aims to improve its agility in the face of frequent changes in the production plan. The simulation steps introduced by Banks et al. [51] were performed rigorously to build a valid simulation model of the real-world factory for further experimentation and optimization. The simulation model is presented in Fig. 6.

The studied company buys its cylinder heads and blocks from an external foundry. When the blocks and cylinder heads arrive at the factory, the raw material goes through a machining area before they are introduced into the simulated assembly line. The assembly line is responsible for the production of three distinct marine engine types, namely D3, D4, and D6, each of which can be customized based on over 300 different specifications. A visual representation of the assembly line can be seen in Fig. 6. The assembly process encompasses 46 operations, including assembly, painting, leakage tests (cold tests), repair, final performance tests (hot tests), customization, and packaging. As depicted in Fig. 6 (a), the production line consists of three assembly areas (Flow 1 to Flow 3), leakage test area, repair area, pre-paint, paint shop, post-paint assembly, hot engine test, customization, and packaging. The transportation between these areas is handled by AFLs.

All operations, except the fully automated paint shop, involve full or partial human intervention. This human element introduces a significant degree of variability into the processes, leading to frequent reworks

Op_1	Op_2	Op_3	…..	Op_{s-1}	Op_s	AFLs
Op_i = Number of Operators in segment i						AFLs

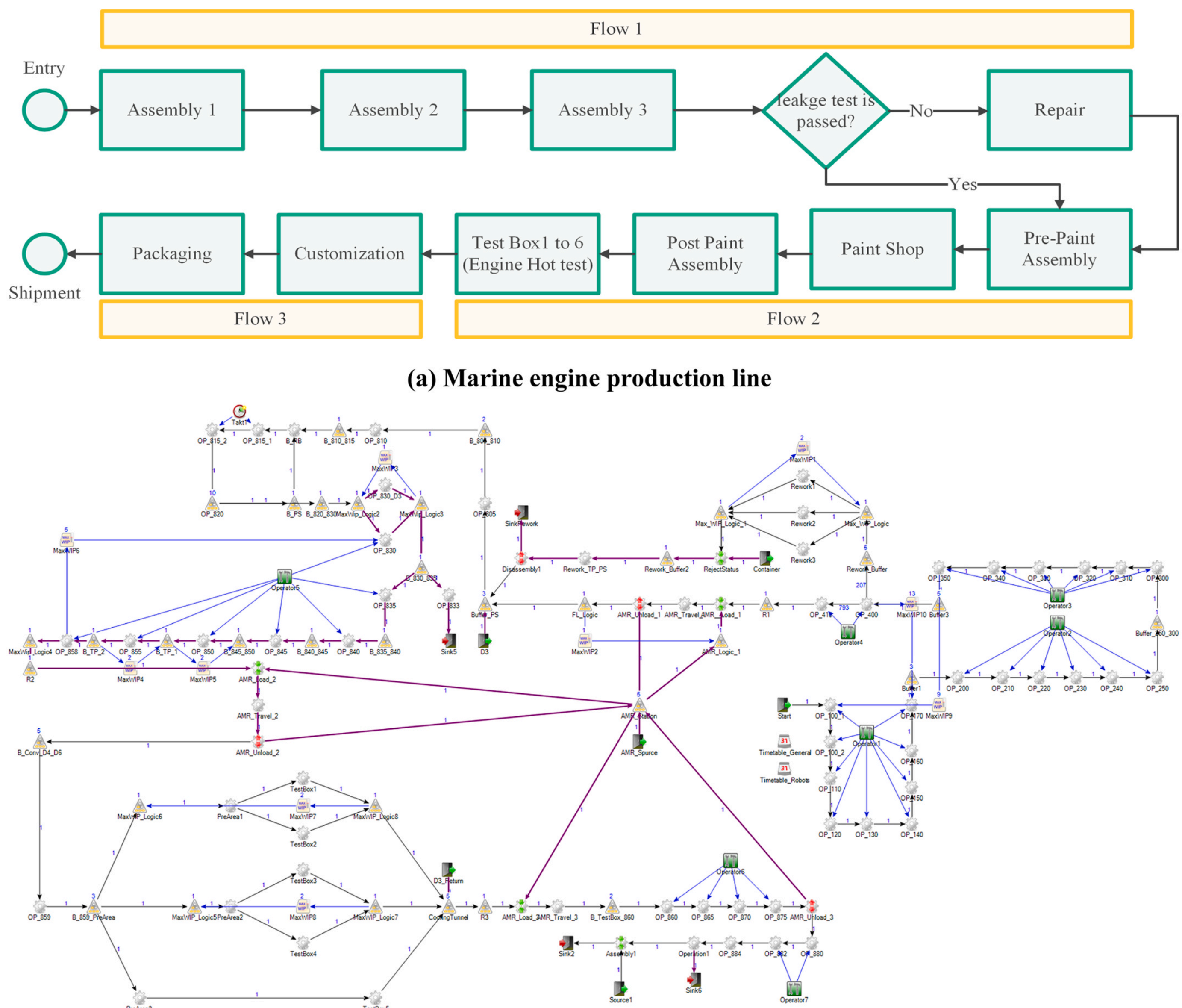
Fig. 5. Solution representation.

and varying cycle times, which are the primary root causes of bottlenecks. On the other hand, the automated paint shop maintains consistent uptime due to a rigorous preventive maintenance regime conducted outside of production hours. This results in high reliability and negligible maintenance-related downtime, a common bottleneck in many production systems. Therefore, in this context, the major driver of bottlenecks is not equipment failure or maintenance-related downtime but the variability in cycle times due to high product variation and manual operations. This underscores the need for optimizing RA to ensure smooth production flow.

The production schedules in the factory are planned three days ahead of the production. The allocation of operators in each area depends on the availability of operators and the specification of engines to be produced in the day. In such a scenario, the manufacturing company's typical objective is maximizing throughput while minimizing the number of operators and AFLs. Given these complexities and the dynamic nature of the production environment, the objective is to provide the decision-makers with a DDS-DSS that is capable of effectively managing the challenges of RA and ensuring production efficiency in the HMLV production environment of the factory.

Fig. 6 (b) represents an image of the simulation model. The model can read the data from the company database by interfacing it with Mircosoft Excel. The data input to the simulation model includes the updated production plan, including engine input sequence and specifications, the availability of the operators for the coming production period, and the availability of AFLs.

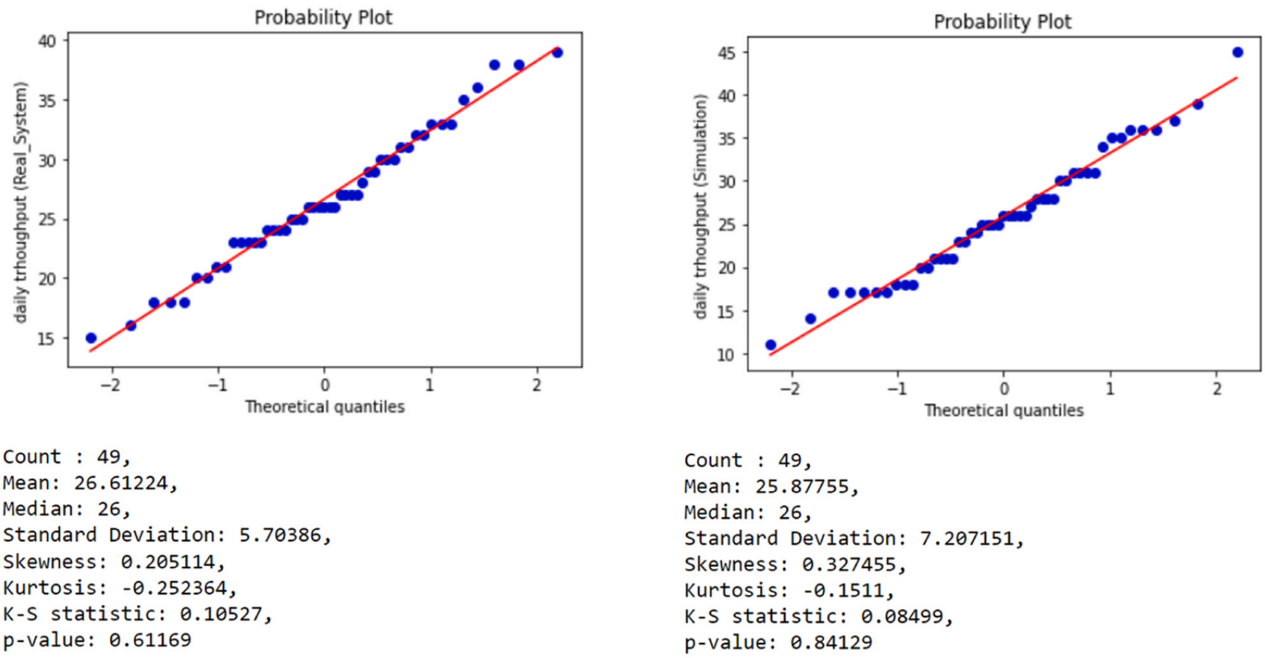
The simulation model was built in a C+ +-based software package, FACTS Analyzer [37], which has proven efficient in simulating various discrete events in the manufacturing context (e.g., [52,53]). Verifying



(a) Marine engine production line

(b) Snapshot of the simulation model in FACTS Analyzer

Fig. 6. A schematic view of the production line and simulation model.



(a) K-Test results for daily throughput of the real system

(b) K-Test results for daily throughput o the simulation model

Fig. 7. Kolmogorov-Smirnov test (K-Test) results.

and validating the model is necessary to ensure the simulation model is qualified for further analysis. The verification of models is an ongoing process that ensures the model is functioning as expected [54].

Model validity was tested using statistical analysis following the validation procedure proposed by [55]. All statistical tests were performed using the Scipy library. As a first step, the Kolmogorov-Smirnov test (K-Test) was used to test the normality of data (i.e., daily engine throughput) for the real system and simulation model. P-values for daily throughput of both the real system and simulation, shown in Fig. 7(a) and (b), are greater than 0.05, indicating that the null hypothesis (i.e., data follows a normal distribution) is not rejected with a 95% confidence interval.

The next step of the validation procedure is natural pairings evaluation. As the data are normally distributed but not paired, the Bartlett test determines whether the variances of real data and simulation results are equal. The p-value calculated for the Bartlett test is equal to 0.11, which is more than 0.05. As a result, at a 95% confidence level, the null hypothesis (i.e., the variances of real system throughput and simulation results are equal) is accepted.

Because the normality of the data and equality of variances were not rejected in the first two steps of the validation procedure, the third step was an independent T-Test. The independent T-Test resulted in a P-value of 0.58. As a result, at a 95% confidence level, the null hypothesis (i.e., the mean values of real system throughput and simulation results are equal) is accepted. According to Chung's Validation Procedure, the simulated model behaves consistently with the actual production line.

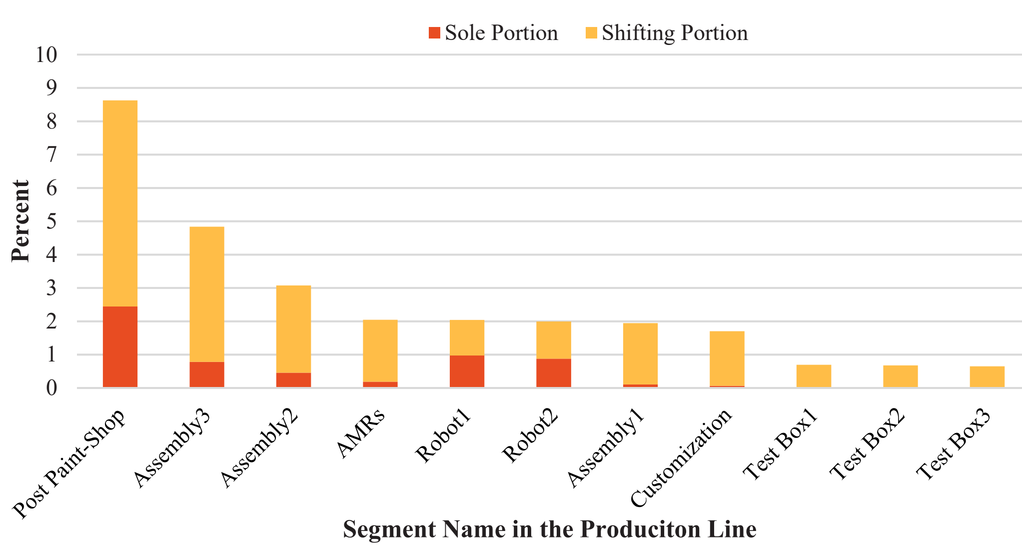


Fig. 8. Bottleneck analysis results using AFLs.

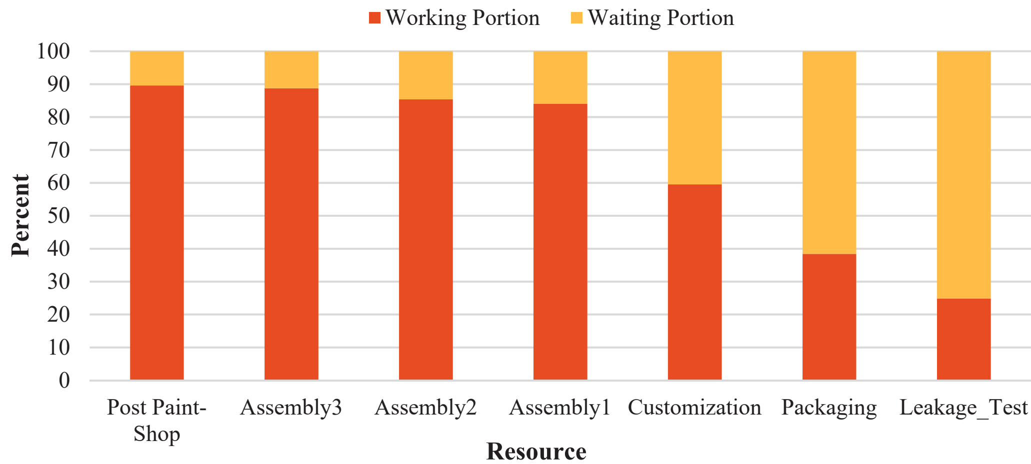


Fig. 9. Utilization of resources in under current scenario.

Therefore, it is concluded that the simulation model is valid.

Results

Scenario analysis using SB-BA

By changing one or several decisions, scenario analysis aims to improve the performance measures of the interest. The number of operators in different sections of the line and the number of available AFLs are the decision variables of the system. Moreover, the performance measures are hourly throughput, the number of operators, and the number of AFLs. Improvement scenarios are designed based on two underlying concepts: the availability and limitations in the allocation of resources. For example, the maximum number of available AFLs and the location of primary bottlenecks in the production process. Fig. 8 illustrates the results of APM based on the simulation output for all operations involved in the production process. As depicted in Fig. 8, APM generates two characteristics for each bottleneck operation, namely, the sole portion and shifting portion, and sorts the operations based on the sum of the sole and shifting portions. According to the results of BA, the workstation following the paint shop is considered the main bottleneck of the production system, followed by Assembly3, Assembly2, AFLs, Robot1, and Robot2.

According to TOC and APM, the first step after detecting the bottleneck is to prevent it from being starved or blocked. Hence, to improve the throughput of the current production line using SB-BA, 68 scenarios are designed according to three types of improvement strategies. The improvement strategies are designed to decrease the number of operators in underutilized segments, relocate them to bottleneck areas, and change the number of AFLs in the production line. Fig. 9 reveals that the customization, packaging, and leakage test area operators are underutilized.

- Strategy 1 includes reducing the number of operators in the underutilized segments (i.e., customization, packaging, and leakage test), which reduces the total number of operators.
- Strategy 2 includes relocating the operators from the under-utilized areas to bottleneck resources. This means that one operator is replaced from the most under-utilized segments of the production line and reassigned to the bottleneck stations. Accordingly, it is desired to evaluate scenarios in which an operator is removed from the under-utilized segments and relocated to Post Paint-Shop, Assembly3, Assembly2, and Assembly1, respectively.
- Strategy 3 includes changing the number of available AFLs. The number of available AFLs can be changed from 3 to 7.

In total, 67 unique improvement scenarios are evaluated and compared regarding throughput, the number of operators, and the number of AFLs. The settings of the production line and the number of unique improvement scenarios under each strategy are presented in the supplementary material (see supplementary material).

The values of the decision variables and the performance measures in each improvement scenario are depicted in Fig. 10 using Parallel Coordinate Plots (PCP). In this figure, the term "_Op" added after the name of each variable represents the number of operators in that segment (e.g., "Assemnly1_Op" is the number of operators in Assemnly1). As shown in Fig. 10 (a), reducing under-utilized resources will decrease the total number of operators in the production line. However, it does not affect the throughput of the production line. Fig. 10 (b) reveals that reallocating under-utilized operators to bottleneck segments can lead to an 11% increase in hourly throughput. According to Fig. 10 (c), decreasing the number of AFLs in production will affect the throughput negatively. However, increasing the number of AFLs cannot significantly improve the throughput.

Although SB-BA resulted in valuable insights about the production line, manually evaluating many possible improvement scenarios is time-consuming. For instance, in this case, it is possible to combine Strategy 1 with either Strategy 2 or Strategy 3, which results in 640 improvement scenarios. As demonstrated in this section, scenario analysis is a viable method for assessing the potential effects of prospective future occurrences on system performance while considering multiple alternative outcomes. Nevertheless, the capacity of scenario analysis to investigate a vast array of possible outcomes is restricted. To put it differently, the scalability of SB-BA presents a significant challenge when dealing with large decision spaces. To overcome this challenge, SB-MOO systematically searches to identify optimal or near-optimal solutions.

SB-MOO

In this segment, after tuning the parameters of the algorithms, first, the results of performing SBO on the real-world case study are presented. Afterwards, the performance of the proposed NSGA-II with meta-learning is compared with NSGA-II and DE. This comparison is based on two most important performance indicators: the number of non-dominated solutions (*NNDS*) and hyper-volume (*HV*) [56,57]. The evaluation is conducted on 30 test problems randomly derived from the original problem. Additionally, we also report the highest throughput discovered by each algorithm.

Parameter tuning

Parameter setting has an important role in the performance of metaheuristic algorithms [58,59]. Thus, conducting prepared experiments to

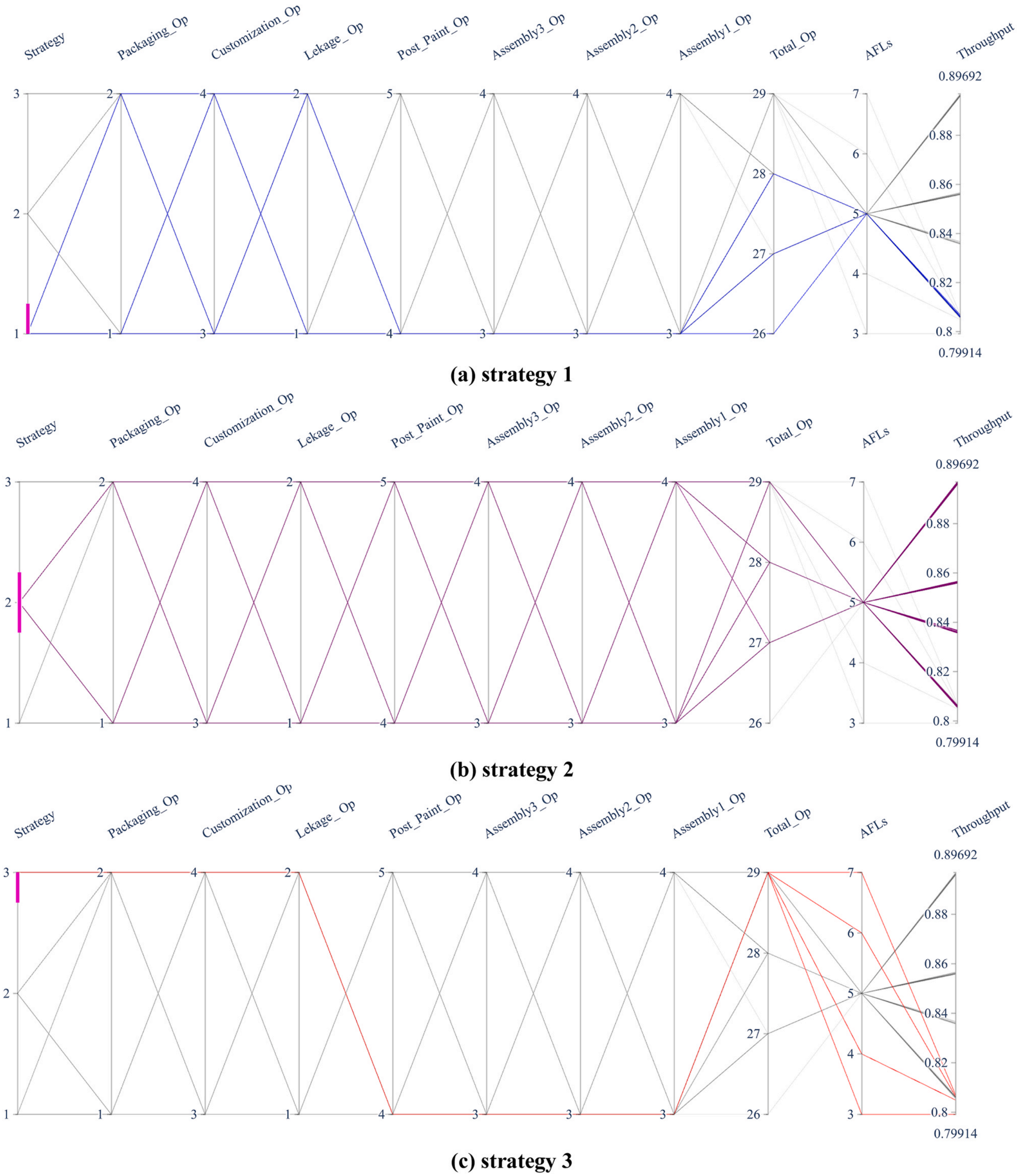


Fig. 10. PCP of the scenarios developed under each strategy.

find a suitable value for each parameter is paramount. The parameters of NSGA-II, NSGA-II-ML, and DE are tuned at four levels using the Gray-based Taguchi method [60], as shown in Table 1. As presented in Table 1, $L_{16}(4^4)$ Taguchi Orthogonal Array was employed to design experiments to tune the parameters of the optimization algorithms. Following the Grey-based Taguchi Method, a reasonable combination of parameter values is set to select the best parameter levels to maximize a certain measure known as the signal-to-noise (S/N) ratio [61]. The main effects diagrams for Means and S/N ratios are plotted in Fig. 11. According to the Taguchi Method, the best levels of the parameters are those falling on the highest S/N levels, resulting in the selected levels for parameters of NSGA-II and NSGA-II-ML to be population size (PS= 100), crossover probability ($P_C = 0.90$), mutation probability ($P_m = 0.3$), and number of evaluations ($NE = 3000$). In the DE case, the selected values are the Differential Weight ($DW = 0.8$), population size (PS= 100), crossover probability ($P_C = 0.70$), mutation probability ($P_m = 0.5$), and number of evaluations ($NE = 5000$).

Table 1
The levels of the parameters for the algorithm.

Algorithm	Parameter	Level_1	Level_2	Level_3	Level_4
NSGA-II and NSGA-II with meta-learning	PS	20	50	100	150
	P_C	0.7	0.8	0.90	0.95
	P_m	0.1	0.15	0.2	0.3
	NE	500	1000	3000	5000
DE	PS	20	50	100	150
	P_C	0.7	0.8	0.9	0.95
	P_m	0.2	0.3	0.4	0.5
	Differential Weight (DW)	0.2	0.4	0.6	0.8
	NE	500	1000	3000	5000

Results of SB-MOO on the application study

The results of SB-MOO, using the proposed NSGA-II-ML, NSGA-II, and DE, are depicted in Fig. 12, which is a 3D scatter plot of the objective space. The highest achieved hourly throughput by NSGA-II-ML, NSGA-II, and DE are 1.01, 0.93, and 0.96, respectively, which is 24%, 16%, and 20% higher than the current throughput of the system. As depicted in Fig. 12, NSGA-II-ML achieves the best throughput among the three algorithms, given a similar number of iterations. This superior performance can be attributed to the meta-learning mechanism inherent in NSGA-II-ML. This meta-learning mechanism allows NSGA-II-ML a better exploration of decision space, as it can leverage the knowledge gained from previous iterations. This is particularly beneficial in the case of large problems, where the search space is vast and the number of potential solutions is enormous.

Performance evaluation

In order to thoroughly assess the performance and scalability of the proposed NSGA-II-ML, it has been benchmarked against NSGA-II and DE over an extensive evaluation across 30 randomly generated instances. These instances are generated with a diverse range of decision space sizes, allowing a careful comparison of the algorithms. The instances are generated in 3 groups; small instances have a decision space of size 10^3 to 10^4; medium instances have a decision space of size 10^5; and large instances have a decision space of size 10^9. Detailed information regarding the generated instances, including the lower and upper bounds of decision variables, as well as the size of the decision space for each instance, can be found in the supplementary material (supplementary material). Fig. 13 presents the performance of NSGA-II-ML, DE, and NSGA-II over two performance measures.

In the analysis of *NNDS* (Fig. 13 (a)) across small-sized, medium-sized, and large-sized problems, all three algorithms - NSGA-II-ML, NSGA-II, and DE - exhibit an upward trend. For small-sized problems, as the problem size increases, NSGA-II-ML and NSGA-II gradually increase their performance in terms of *NNDS*, with NSGA-II-ML eventually surpassing the others. DE starts strong but remains stable. In the medium-sized problem phase, all algorithms significantly increase their performance, i.e., *NNDS*, adapting to the problem size, with NSGA-II-ML maintaining the lead and DE showing substantial improvement. For large problems, while all algorithms continue to increase their solutions, the rate of increase slows for NSGA-II and DE. NSGA-II-ML, however, maintains a steady growth rate, ending with the highest number of non-dominated solutions.

In the analysis of HV (Fig. 13 (b)) for small-sized, medium-sized, and large-sized problems, all three algorithms - NSGA-II-ML, NSGA-II, and

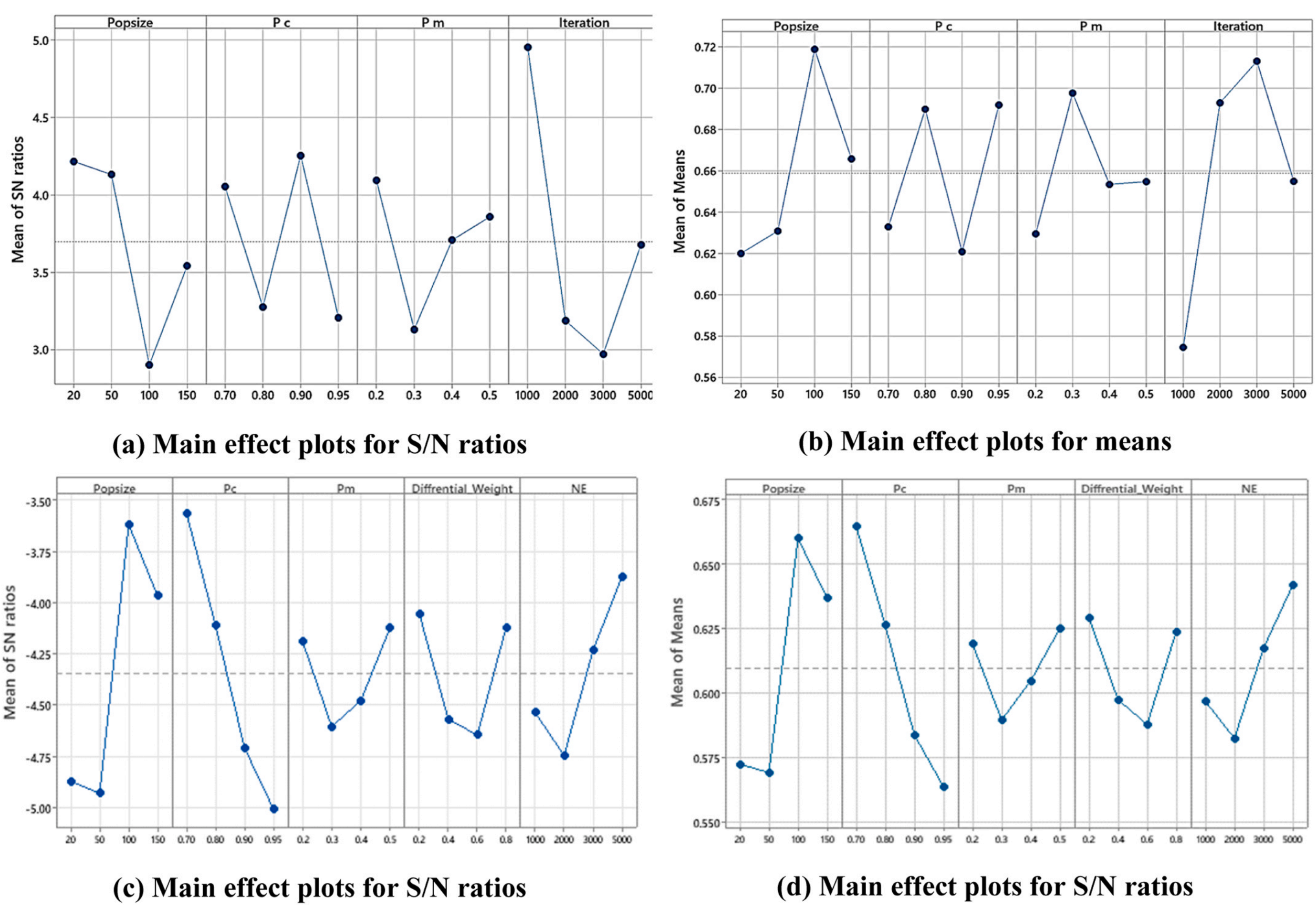


Fig. 11. Parameter tuning of NSGA-II and NSGA-II with meta-learning (a and b) and DE (c and d) using the grey-based Taguchi method.

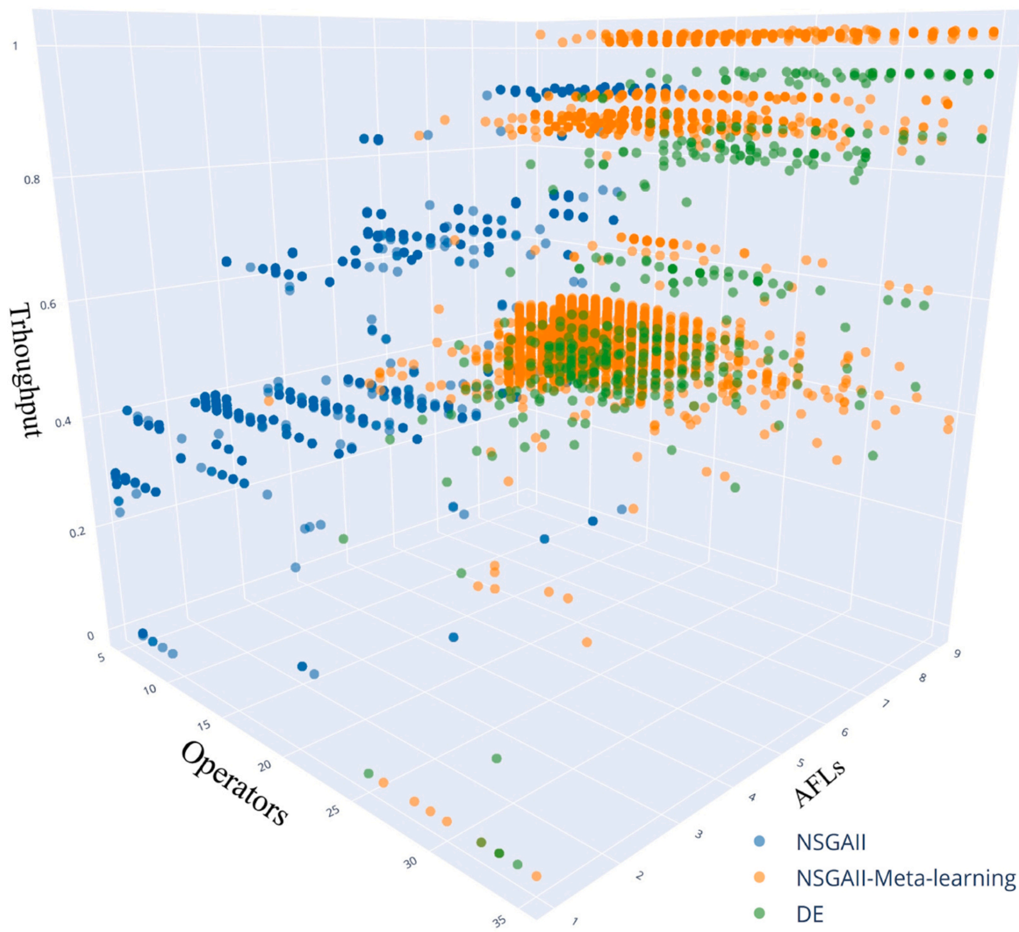


Fig. 12. The 3D-objective space of the optimization performed on the application study by NSGA-II-ML, NSGA-II, and DE.

DE - show an increase as the problem size grows. For small-sized problems, NSGA-II-ML and NSGA-II gradually increase their hypervolume, with NSGA-II-ML eventually surpassing the others, while DE starts strong but remains stable. In the medium-sized problem phase, all algorithms significantly increase their HV, adapting to the problem size, with NSGA-II-ML maintaining the lead and DE showing substantial improvement. For large-sized problems, while all algorithms continue to increase their HV, the rate of increase slows for NSGA-II and DE. NSGA-II-ML, however, maintains a steady growth rate, ending with the highest HV.

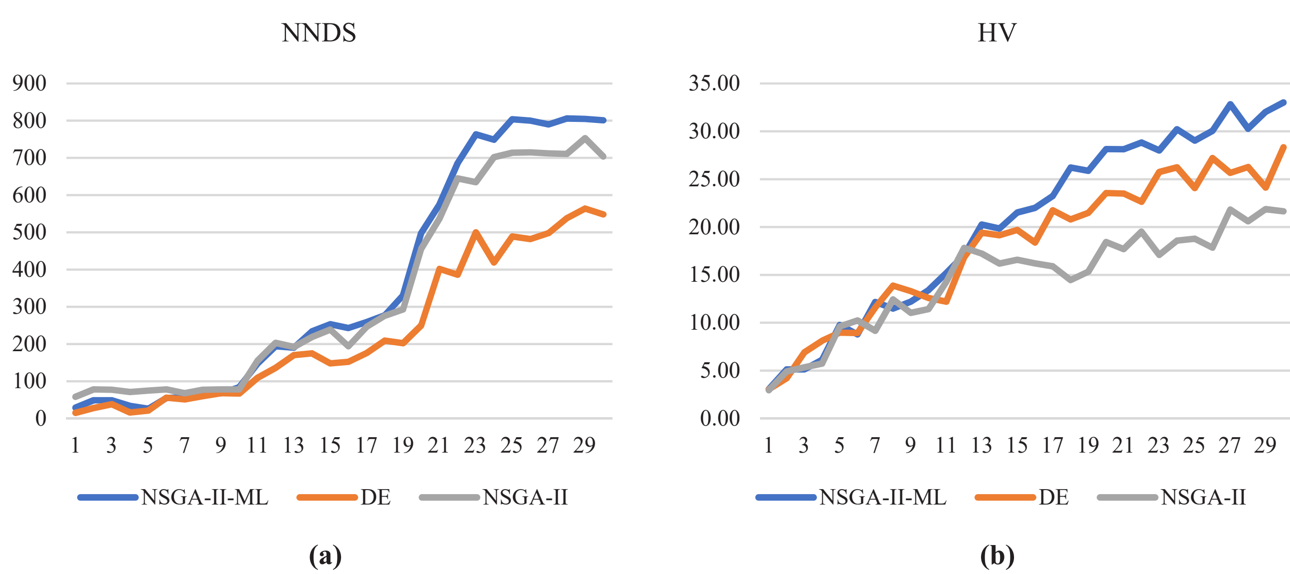


Fig. 13. Performance of NSGA-II-ML, DE, and NSGA-II in two performance measures.

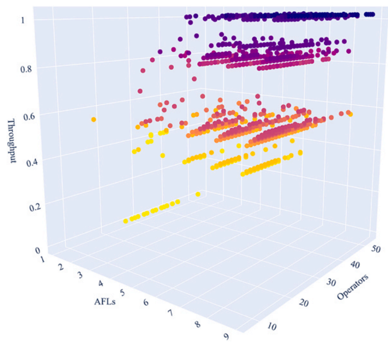


Fig. 14. Results of DBSCAN performed on the objective space of SB-MOO.

Post-optimality analysis

As illustrated in Fig. 14, the DBSCAN algorithm divides the objective space into seven clusters based on throughput values. The rationale behind clustering based on the throughput values is rooted in the mobility of operators and AFLs, as reallocating these resources does not entail substantial overhead costs to existing operations. Consequently, this study has aimed to identify the rules governing the cluster of solutions situated in the upper-right corner of Fig. 14. These solutions exhibit high values for throughput. The presence of multiple solutions within the cluster indicates the necessity of conducting statistical analysis to determine the range of decision variables that lead to the optimal region of the objective space.

As it is visually clear in Fig. 14, it has demonstrated remarkable efficacy in clustering the results of MOO. The effectiveness of this method is predominantly attributed to its distinctive capability to detect clusters with diverse shapes and dimensions.

Analyzing the solutions falling in the desired cluster(s) can provide managers with insightful knowledge about the variable bounds that can guide the production towards that cluster. In this study, a boxplot is used to perform frequency analysis and accordingly extract the values of decision variables that can result in the highest throughput values. As depicted in Fig. 15, the total number of operators for the solutions with the best achievable throughput falls in the range of 26–29. The number of AFLs can be 5 or 6. The values for the number of operators in each segment are also evident in Fig. 15.

Managerial insights

The following points critically analyze the results of the current study and propose managerial insights for production systems.

SB-BA has proven to be a powerful and intuitive tool for handling the RA problem in HMLV production systems. However, its capacity to explore large decision spaces is somewhat limited. This limitation becomes more pronounced in scenarios with larger decision spaces, where SB-MOO proves to be a more suitable option. For instance, the industrial case studied in this paper operates in two distinct modes: a two-shift mode and a one-shift mode. In the two-shift mode, the decision space for each shift is smaller due to the limited number of operators, making it manageable for SB-BA. However, in the one-shift mode, where all operators are consolidated into a single shift, the number of operators is higher, leading to a larger decision space. This expansion of the decision space exposes the limitations of SB-BA's capacity to handle such scenarios.

The second point pertains to applying meta-learning within stochastic short-term RA problems. Meta-learning, as an adaptive modification approach, has been shown to improve the algorithm's performance in terms of scalability and exploration of the decision space. It allows the algorithm to learn from evolving populations and adjust its parameters in response, thereby enhancing its performance over time. This novel application of meta-learning to the NSGA-II within the context of RA potentially offers a more efficient and effective solution to complex, dynamic RA problems.

Moreover, the results of this study indicated that the reallocation of mobile resources could substantially impact throughput, which presents a promising opportunity for enhancing operational efficiency. Nevertheless, this procedure presents specific difficulties, especially regarding the redistribution of human resources within a manufacturing system. From a cognitive standpoint, it is imperative to comprehend the effects of frequent reallocation on operators who are frequently subjected to such adaptations. One cannot ignore the possibility of unintended consequences that might have a long- or medium-term negative impact on the employee's overall performance. An employee who experiences frequent task switching may encounter difficulties in coping with the intricacies of the transitions, resulting in a deterioration of work output and a rise in occupational mistakes.

Hence, it is imperative for managers to diligently oversee the circumstances and carefully observe any alterations in a worker's behavior

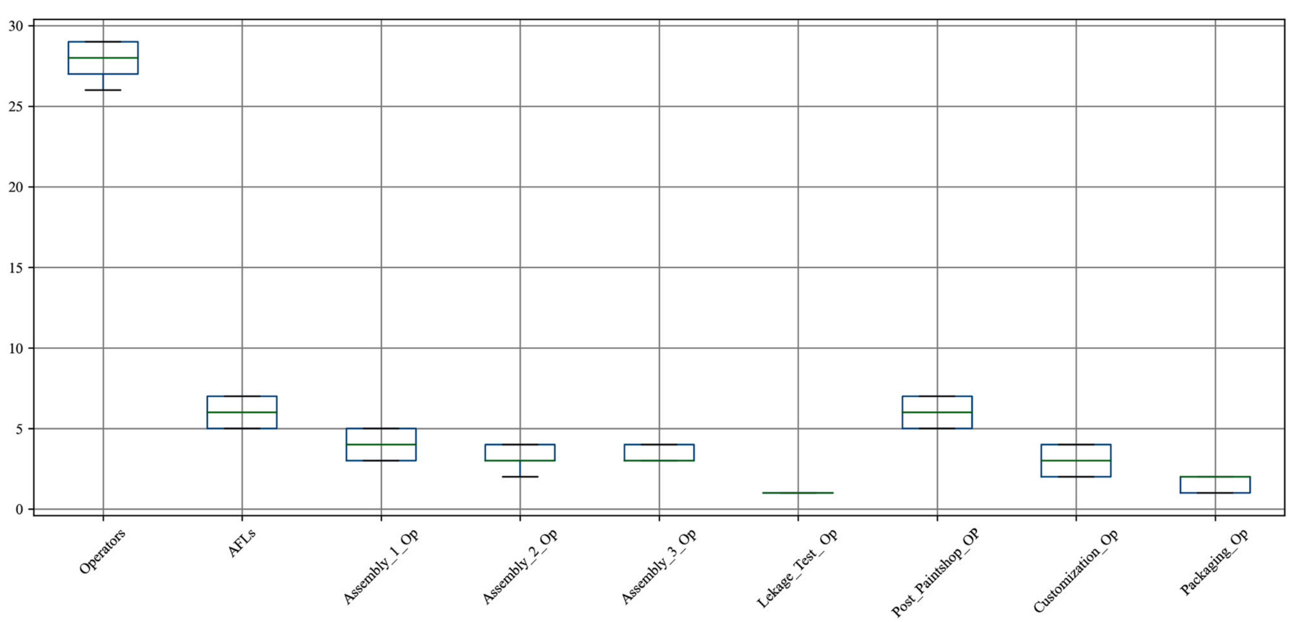


Fig. 15. Boxplot of optimal decision variables for maximum throughput.

or attitude. With this kind of monitoring, issues can be found early and fixed immediately. In addition, it is imperative to furnish the employees with the necessary assistance and instruction to facilitate their adjustment to the modifications. Potential strategies to enhance employee performance may include providing supplementary training sessions, furnishing unambiguous instructions for new assignments, or instituting a mentorship initiative whereby experienced workers can guide inexperienced operators. By following these guidelines, managers can ensure a smooth transition during reallocation, preserving productivity and morale while lowering the risk of mistakes.

Conclusions and future research

This paper has addressed one of the significant challenges of HMLV manufacturing systems, i.e., appropriate allocation of resources, by focusing on providing a DDS-DSS. The proposed DDS-DSS incorporates a simulation model capable of receiving updated data, including, but not limited to updates in received orders, availability of resources, workstation processing times, machine failures, product rejection rates in quality inspection, etc. The updates could happen on a pre-determined time interval or user command. This system allows resource allocation using two powerful approaches, SB-BA and SB-MOO, with the method being dependent on the decision-maker's preferences. The SB-BA benefits from a recently developed data-driven BA approach. Then, different RA strategies, i.e., reducing the number of operators in under-utilized segments, relocating the operators from under-utilized areas to bottlenecks, and changing the number of available AFLs, are designed based on BA results to find the near-optimal allocation of mobile resources, namely, operators and AFLs.

SB-MOO utilizes a popular MOO algorithm, namely, NSGA-II, that has been demonstrated to be an effective approach to performing RA in response to changes in demand and production plans. Additionally, DBSCAN is employed to cluster the MOO results for post-optimality analysis. The results revealed that general rules of thumb could be extracted to allocate resources to achieve the highest possible throughput.

While the SB-BA approach was found to be more limited in its exploration capabilities, it was also more controllable and intuitive, making it a suitable choice for decision-makers who prefer a more straightforward approach. On the other hand, despite requiring certain knowledge and skill, SB-MOO has been proved to be a powerful tool for RA in today's volatile environment.

In this study, the SB-MOO approach has been further improved using a meta-learning mechanism to enhance the performance of NSGA-II-ML. The comparison between NSGA-II-ML, NSGA-II, and Differential Evolution (DE) algorithms shows that NSGA-II-ML demonstrates superior scalability, consistently improving its performance as problem sizes increase. It consistently outperforms the other two algorithms across all problem sizes, achieving the highest number of non-dominated solutions and the highest hypervolume. This superior performance can be attributed to the meta-learning mechanism inherent in NSGA-II-ML, which allows it to adapt more quickly to new problems or larger problem sizes by leveraging the knowledge gained from previous iterations.

Overall, this study underscores the potential of DDS-DSS as a valuable tool in managing and optimizing HMLV production systems, offering decision-makers a flexible and effective approach to RA. The results revealed that an 11% improvement is attainable using the SB-BA through analyzing a limited number of scenarios. Moreover, the optimization results obtained by NSGA-II-ML, NSGA-II, and DE showed 24%, 16%, and 20% higher hourly throughput than the current throughput of the system, respectively. These findings highlight the power of the improved SB-MOO approach, particularly in today's volatile environment, and the benefits of integrating meta-learning mechanisms into MOO algorithms for resource allocation in HMLV production systems. The results derived from the post-optimality analysis indicate that the optimal solutions, which yield the highest achievable throughput, necessitate a total operator count ranging between 26 and 29. Additionally, the quantity of AFLs required for these solutions can either be 5 or 6. Overall, this study underscores the potential of DDS-DSS as a valuable tool in managing and optimizing HMLV production systems, offering decision-makers a flexible and effective approach to RA.

There are two promising directions for future research within the domain of DDS-DSS in manufacturing systems, expanding upon the findings of this study. The first direction could concentrate on broadening DDS-DSS application across different manufacturing sectors. In contrast, the second direction could focus on enhancing the performance of DDS-DSS as a promising tool for improving manufacturing systems.

Regarding the first research direction, the application of DDS-DSS could be extended beyond RA to include a broader spectrum of problems prevalent in the manufacturing industry, potentially leading to more substantial improvements. Various demand-dependent decision-making problems could be considered in conjunction with RA. One of these problems is the buffer allocation problem, which involves determining the optimal locations and sizes of buffers to maximize throughput and minimize work-in-process. Additionally, advanced demand-driven logistics management approaches, including demand-driven material requirement planning, could also benefit from the application of DDS-DSS. By integrating DDS-DSS with production planning, allocation of resources and buffers, and logistics management, manufacturers could more accurately forecast demand and plan their production, thereby reducing inventory costs and improving customer service. Moreover, expanding the DSS method to additional complex line topologies and controls, e.g., fully dynamic moving assembly systems, could enable even greater productivity improvements through real-time resource allocation.

The second research direction in DDS-DSS could focus on improving the internal structure and performance of DDS-DSS by developing more robust and efficient algorithms capable of handling large-scale complex manufacturing systems. This research direction focuses on developing more advanced data analytics techniques to extract and manage the vast data generated in manufacturing processes. Future research could also explore the use of more sophisticated or alternative algorithms, potentially enhancing the efficiency and effectiveness of the optimization procedure. Moreover, integrating machine learning and artificial intelligence in DDS-DSS could significantly enhance decision-making. For instance, integrating data-driven predictive models (e.g., the study by Subramaniyan et al. [22]) with a simulation model for optimal resource allocation could be a compelling exploration area for practitioners and researchers. While this approach promises to enhance bottleneck prediction, it also presents challenges such as dependency on the range and diversity of training data and increased computational complexity. In this regard, future studies should focus on developing robust models capable of handling significant deviations from historical production situations. Additionally, strategies to manage the computational efficiency of the integrated system, possibly through parallel processing or model simplification, should be investigated. Furthermore, the production line studied in this research represents a system with relatively low product flow velocity, as is common in industries dealing with physically large and complex products. In such cases, bottlenecks tend to be more persistent than frequently shifting. As the industrial partner indicated, the primary challenge is mitigating the severity of these bottlenecks, which was the focus of this work. However, a key strength of the DDS-DSS is that the simulation model can be continuously updated based on real-time data to adapt to emerging disruptions or changes in the production line. This makes it possible to use the DDS-DSS in production systems with higher bottleneck shiftiness. However, the higher frequency of updating and running DDS-DSS requires more processing power. Hence, enhancing the capability of the proposed approach to address shifting bottlenecks in production systems with greater variability and dynamics is an important direction for future research.

Alongside the focus on DES for decision support in this study, the growing interest in Artificial Intelligence (AI) based decision support systems in production system resource management should be acknowledged, as evidenced by Subramaniyan et al. [62]. Combining DES and AI-based decision support could offer a more comprehensive and effective approach to managing production system resources. This integration could potentially enhance the prediction and management of bottlenecks, providing a more dynamic and adaptable framework for resource allocation. Therefore, future research should explore this integration to enrich the discourse further and provide more robust solutions for production system management.

Funding

This study was funded by the Knowledge Foundation (KKS) and Sweden's Innovation Agency via the ACCURATE 4.0 (grant agreement No. 20200181) and PREFER projects, respectively.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to express their gratitude to their industrial partner, Volvo Penta of Sweden, for their collaborative support throughout the project.

References

[1] Choi TM, Kumar S, Yue X, Chan H-L. Disruptive technologies and operations management in the industry 4.0 era and beyond. Prod Oper Manag 2022;31:9–31. https://doi.org/10.1111/poms.13622.

[2] Ghobakhloo M, Fathi M, Iranmanesh M, Maroufkhani P, Morales ME. Industry 4.0 ten years on: a bibliometric and systematic review of concepts, sustainability value drivers, and success determinants. J Clean Prod 2021;302:127052. https://doi.org/10.1016/j.jclepro.2021.127052.

[3] Enrique DV, Marodin GA, Santos FBC, Frank AG. Implementing industry 4.0 for flexibility, quality, and productivity improvement: technology arrangements for different purposes. Int J Prod Res 2022;0:1–26. https://doi.org/10.1080/00207543.2022.2142689.

[4] Ghobakhloo M, Fathi M. Industry 4.0 and opportunities for energy sustainability. J Clean Prod 2021;295:126427. https://doi.org/10.1016/j.jclepro.2021.126427.

[5] Raj A, Jeyaraj A. Antecedents and consequents of industry 4.0 adoption using technology, organization and environment (TOE) framework: a meta-analysis. Ann Oper Res 2023;322:101–24. https://doi.org/10.1007/s10479-022-04942-7.

[6] de Paula Ferreira W, Armellini F, De Santa-Eulalia LA. Simulation in industry 4.0: a state-of-the-art review. Comput Ind Eng 2020;149:106868.

[7] Goodall P, Sharpe R, West A. A data-driven simulation to support remanufacturing operations. Comput Ind 2019;105:48–60. https://doi.org/10.1016/j.compind.2018.11.001.

[8] Saez M, Barton K, Maturana F, Tilbury DMM. Modeling framework to support decision making and control of manufacturing systems considering the relationship between productivity, reliability, quality, and energy consumption. J Manuf Syst 2022;62:925–38. https://doi.org/10.1016/j.jmsy.2021.03.011.

[9] Tseng M-L, Tran TPT, Ha HM, Bui T-D, Lim MK. Sustainable industrial and operation engineering trends and challenges toward industry 4.0: a data driven analysis. J Ind Prod Eng 2021;38:581–98. https://doi.org/10.1080/21681015.2021.1950227.

[10] Henrique dos Santos C, Montevechi JAB, de Queiroz JA, de Carvalho Miranda R, Leal F. Decision support in productive processes through DES and ABS in the digital twin era: a systematic literature review. Int J Prod Res 2022;60:2662–81. https://doi.org/10.1080/00207543.2021.1898691.

[11] Chinnathai MK, Alkan B, Harrison R. A novel data-driven approach to support decision-making during production scale-up of assembly systems. J Manuf Syst 2021;59:577–95. https://doi.org/10.1016/j.jmsy.2021.03.018.

[12] Nabi HZ, Aized T, Riaz F. Modeling, analysis and optimization of carousel-based flexible manufacturing system. J Ind Prod Eng 2022;39:479–93. https://doi.org/10.1080/21681015.2022.2031323.

[13] Saint-Guillain M, Gibaszek J, Vaquero T, Chien S. Romie: a domain-independent tool for computer-aided robust operations management. Eng Appl Artif Intell 2022;111:104801. https://doi.org/10.1016/j.engappai.2022.104801.

[14] Santos MS, Pinto TVB, Júnior ÊL, Cota LP, Souza MJF, Euzébio TAM. Simheuristic-based decision support system for efficiency improvement of an iron ore crusher circuit. Eng Appl Artif Intell 2020;94:103789. https://doi.org/10.1016/j.engappai.2020.103789.

[15] Zhou B, Bao J, Li J, Lu Y, Liu T, Zhang Q. A novel knowledge graph-based optimization approach for resource allocation in discrete manufacturing workshops. Robot Comput-Integr Manuf 2021;71:102160. https://doi.org/10.1016/j.rcim.2021.102160.

[16] Bouajaja S, Dridi N. A survey on human resource allocation problem and its applications. Oper Res 2017;17:339–69. https://doi.org/10.1007/s12351-016-0247-8.

[17] Mendonça PA, da Piedade Francisco R, de Souza Rabelo D. OEE approach applied to additive manufacturing systems in distributed manufacturing networks. Comput Ind Eng 2022;171:108359. https://doi.org/10.1016/j.cie.2022.108359.

[18] Chiu C-C, Lin JT. An efficient elite-based simulation–optimization approach for stochastic resource allocation problems in manufacturing and service systems. Asia-Pac J Oper Res 2022;39:2150030. https://doi.org/10.1142/S0217595921500305.

[19] Schmenner RW. The pursuit of productivity. Prod Oper Manag 2015;24:341–50. https://doi.org/10.1111/POMS.12230.

[20] Mahmoodi E, Fathi M, Ghobakhloo M. The impact of Industry 4.0 on bottleneck analysis in production and manufacturing: current trends and future perspectives. Comput Ind Eng 2022;174:108801. https://doi.org/10.1016/j.cie.2022.108801.

[21] Latsou C, Farsi M, Erkoyuncu JA. Digital twin-enabled automated anomaly detection and bottleneck identification in complex manufacturing systems using a multi-agent approach. J Manuf Syst 2023;67:242–64. https://doi.org/10.1016/j.jmsy.2023.02.008.

[22] Subramaniyan M, Skoogh A, Sheikh Muhammad A, Bokrantz J, Turanoğlu Bekar E. A prognostic algorithm to prescribe improvement measures on throughput bottlenecks. J Manuf Syst 2019;53:271–81. https://doi.org/10.1016/j.jmsy.2019.07.004.

[23] Debevec M, Simic M, Jovanovic V, Herakovic N. Virtual factory as a useful tool for improving production processes. J Manuf Syst 2020;57:379–89. https://doi.org/10.1016/j.jmsy.2020.10.018.

[24] Lee J, Yang C. Deep neural network and meta-learning-based reactive sputtering with small data sample counts. J Manuf Syst 2022;62:703–17.

[25] Li L, Wang Y, Xu Y, Lin K. Meta-learning based industrial intelligence of feature nearest algorithm selection framework for classification problems. J Manuf Syst 2022;62:767–76.

[26] Wang D, Liu Q, Wu D, Wang L. Meta domain generalization for smart manufacturing: tool wear prediction with small data. J Manuf Syst 2022;62:441–9.

[27] Hevner AR, March ST, Park J, Ram S. Design science in information systems research. MIS Q Manag Inf Syst 2004;28:75–105.

[28] Pierce N.G., Yurtsever T. Dynamic dispatch and graphical monitoring system. In: Proceedings of the 1999 IEEE International Symposium on Semiconductor Manufacturing Conference Proceedings (Cat No.99CH36314).1999. pp. 65–68.

[29] Koyuncu N., Lee S., Sarfare P., Son Y.J. Dynamic-data-driven adaptive multi-scale simulation (DDDAMS) for planning and control of distributed manufacturing enterprises: In: Proceedings of the IIE Annual Conference and Expo 2008.2008. pp. 1611–1616. Available: ⟨http://www.scopus.com/inward/record.url?scp=63849186363&partnerID=8YFLogxK⟩. Accessed 23 Jun 2023.

[30] Segura Velandia DM, West AA, Conway PP. A database system for decision support in low-volume electronics assembly. Proc Inst Mech Eng Part B J Eng Manuf 2011;225:1411–30. https://doi.org/10.1177/2041297510393566.

[31] Hussaini S, Lahrman G. Using modeling and simulation in the manufacturing of medical devices. Adv Mater Res 2012;488–489:1032–8. https://doi.org/10.4028/www.scientific.net/AMR.488-489.1032.

[32] Rashid K, Louis J, Swanson C. Optimizing labor allocation in modular construction factory using discrete event simulation and genetic algorithm. Winter Simul Conf (WSC) 2020;2020:2569–76.

[33] Wang X, Liu Y, He Y, Li N, Mu H, Bai Y. Achieving manufacturing excellence through the integration of process planning change and data-driven simulation. J Phys Conf Ser 2020;1693:012045. https://doi.org/10.1088/1742-6596/1693/1/012045.

[34] Fani V, Bindi B, Bandinelli R. Designing and optimizing production in a high variety / low volume environment through data-driven simulation. ECMS 2021 Proceedings edited by Khalid Al-Begain, Mauro Iacono, Lelio Campanile, Andrzej Bargiela. ECMS,; 2021. p. 10–5. ⟨10.7148/2021-0010⟩.

[35] Luo D, Guan Z, He C, Gong Y, Yue L. Data-driven cloud simulation architecture for automated flexible production lines: application in real smart factories. Int J Prod Res 2022;60:3751–73. https://doi.org/10.1080/00207543.2021.1931977.

[36] Sakr AH, Aboelhassan A, Yacout S, Bassetto S. Simulation and deep reinforcement learning for adaptive dispatching in semiconductor manufacturing systems. J Intell Manuf 2023;34:1311–24. https://doi.org/10.1007/s10845-021-01851-7.

[37] Ng A., Bernedixen J. Production systems analysis and optimization using facts analyzer. In: Proceedings of the 2018 Winter Simulation Conference.2018. Available: ⟨https://dl.acm.org/doi/abs/10.5555/3320516.3321120⟩. Accessed 27 Jun 2022.

[38] Cavana RY, Dangerfield BC, Pavlov OV, Radzicki MJ, Wheat ID, editors. Feedback Economics: Economic Modeling with System Dynamics. Cham: Springer International Publishing; 2021. ⟨10.1007/978-3-030-67190-7⟩.

[39] Kumbhar M, Ng AHC, Bandaru S. A digital twin based framework for detection, diagnosis, and improvement of throughput bottlenecks. J Manuf Syst 2023;66:92–106. https://doi.org/10.1016/j.jmsy.2022.11.016.

[40] Roser C, Lorentzen K, Lenze D, Deuse J, Klenner F, Richter R, et al. Bottleneck prediction using the active period method in combination with buffer inventories. IFIP Adv Inf Commun Technol 2017;514:374–81. https://doi.org/10.1007/978-3-319-66926-7_43.

[41] Mahmoodi E. APM (1.1). 2023. doi:10.5281/zenodo.8140390.

[42] Verma S, Pant M, Snasel V. A comprehensive review on NSGA-II for multi-objective combinatorial optimization problems. IEEE Access 2021;9:57757–91.
[43] Nourmohammadi A, Fathi M, Ng AHC. Balancing and scheduling assembly lines with human-robot collaboration tasks. Comput Oper Res 2022;140:105674. https://doi.org/10.1016/j.cor.2021.105674.
[44] Yue C, Suganthan PN, Liang J, Qu B, Yu K, Zhu Y, et al. Differential evolution using improved crowding distance for multimodal multiobjective optimization. Swarm Evol Comput 2021;62:100849. https://doi.org/10.1016/j.swevo.2021.100849.
[45] Salem M, Mora AM, Guervos JJM. Overtaking uncertainty with evolutionary TORCS controllers: combining BLX with decreasing α operator and grand prix selection. IEEE Trans Games 2022;14:318–27.
[46] Singh G, Gupta N. A study of crossover operators in genetic algorithms. In: Khosravy M, Gupta N, Patel N, editors. Frontiers in Nature-Inspired Industrial Optimization. Springer Tracts in Nature-Inspired Computing. Singapore: Springer; 2022. p. 17–32. ⟨10.1007/978-981-16-3128-3_2⟩.
[47] Zhang Z., Lu B. Improving NSGA-II by a Local Search Strategy with Gaussian Mutation. In: Proceedings of the 2021 40th Chinese Control Conference (CCC).2021. pp. 1628–1633.
[48] Bandaru S, Ng AHC, Deb K. Data mining methods for knowledge discovery in multi-objective optimization: Part B - New developments and applications. Expert Syst Appl 2017;70:119–38.
[49] Sahraei S, Asadzadeh M. Cluster-based multi-objective optimization for identifying diverse design options: application to water resources problems. Environ Model Softw 2021;135:104902. https://doi.org/10.1016/j.envsoft.2020.104902.
[50] Chiu M-C, Lee Y-H, Chen T-M. Integrating content-based image retrieval and deep learning to improve wafer bin map defect patterns classification. J Ind Prod Eng 2022;39:614–28. https://doi.org/10.1080/21681015.2022.2074155.
[51] Banks J, Carson JS, Nelson BL, Nicol DM. Discrete-Event System Simulation. Prentice Hall,; 2013.
[52] Zúñiga ER, Moris MU, Syberfeldt A, Fathi M, Rubio-Romero JC. A simulation-based optimization methodology for facility layout design in manufacturing. IEEE Access 2020;8:163818–28. https://doi.org/10.1109/ACCESS.2020.3021753.
[53] Ruiz Zúñiga E, Flores García E, Urenda Moris M, Fathi M, Syberfeldt A. Holistic simulation-based optimisation methodology for facility layout design with consideration to production and logistics constraints. Proc Inst Mech Eng Part B J Eng Manuf 2021;235:2350–61. https://doi.org/10.1177/09544054211017310.
[54] Sargent RG, Balci O. History of verification and validation of simulation models. Winter Simul Conf (WSC) 2017;2017:292–307.
[55] Chung CA. Simulation Modelling Handbook: A Practical Approach. CRC Press,; 2004.
[56] Khettabi I, Benyoucef L, Boutiche MA. Sustainable reconfigurable manufacturing system design using adapted multi-objective evolutionary-based approaches. Int J Adv Manuf Technol 2021;115:3741–59. https://doi.org/10.1007/s00170-021-07337-3.
[57] Nourmohammadi A, Ng AHC, Fathi M, Vollebregt J, Hanson L. Multi-objective optimization of mixed-model assembly lines incorporating musculoskeletal risks assessment using digital human modeling. CIRP J Manuf Sci Technol 2023;47: 71–85. https://doi.org/10.1016/j.cirpj.2023.09.002.
[58] Beheshtinia MA, Jozi A, Fathi M. Optimizing disaster relief goods distribution and transportation: a mathematical model and metaheuristic algorithms. Appl Math Sci Eng 2023;31:2252980. https://doi.org/10.1080/27690911.2023.2252980.
[59] Beheshtinia MA, Fathi M. Energy-efficient and sustainable supply chain in the manufacturing industry. Energy Sci Eng 2023;11:357–82. https://doi.org/10.1002/ese3.1337.
[60] Kuo Y, Yang T, Huang GW. The use of a grey-based Taguchi method for optimizing multi-response simulation problems. Eng Optim 2008;40:517–28. https://doi.org/10.1080/03052150701857645.
[61] Taguchi G, Chowdhury Subir, Wu Yuin, Taguchi Shin, Yano H. Taguchi's Quality Engineering Handbook. John Wiley & Sons,; 2005.
[62] Subramaniyan M, Skoogh A, Bokrantz J, Sheikh MA, Thürer M, Chang Q. Artificial intelligence for throughput bottleneck analysis – state-of-the-art and future directions. J Manuf Syst 2021;60:734–51.