*Sci*

MDPI

*Article*

# A Feasibility-Oriented Heuristic for Garment Assembly Line Balancing with Machine Compatibility Constraints

**Yasmine El Belghiti [1,2,*], Abdelfattah Mouloud [2,*] and Aziz Soulhi [1]**

[1] National Higher School of Mines of Rabat (ENSMR), Avenue Hadj Ahmed Cherkaoui, Agdal, P.O. Box 753, Rabat 10000, Morocco; soulhi@enim.ac.ma
[2] Higher School of Textile and Clothing Industries (ESITH), Km 8, Route d'El Jadida, Laymoune, P.O. Box 7731, Casablanca 20200, Morocco
* Correspondence: yasmineelbelghiti2000@gmail.com (Y.E.B.); mouloud@esith.ac.ma (A.M.)

**Abstract**

Garment assembly line balancing requires workstation assignments that satisfy precedence, cycle-time, and machine-compatibility constraints. However, conventional priority-based heuristics may generate numerically attractive but technologically infeasible configurations. This study proposes a Feasibility-Oriented Heuristic (FOH) combining deterministic topological task ordering with incremental verification of workstation capacity and admissible machine types. The procedure preserves the modeled feasibility requirements throughout workstation construction under explicitly defined assumptions. Its performance is evaluated on a 17-operation garment assembly benchmark with a total work content of 20 min and a nominal cycle time of two minutes. Comparisons are conducted against conventional Ranked Positional Weight, Shortest Processing Time, and Largest Candidate Rule heuristics, their machine-compatibility-aware variants, and an exact Mixed-Integer Linear Programming formulation. Under nominal conditions, FOH generates 15 feasible workstations with a line efficiency of 66.67 percent, matching the proven MILP optimum. Conventional RPW and LCR produce machine-incompatible configurations despite achieving higher apparent efficiencies. Across nine deterministic sensitivity scenarios, FOH and all compatibility-aware variants maintain feasibility. FOH matches the optimal workstation count in three scenarios, compared with eight for the compatibility-aware Largest Candidate Rule, and its maximum optimality gap reaches 16.67 percent. The findings establish FOH as a transparent and reproducible feasibility-oriented constructive procedure, while demonstrating that workstation-count performance remains dependent on task-selection strategies. Further validation on larger and independent industrial benchmarks is required to assess its generalizability.

**Keywords:** garment assembly line balancing; feasibility-oriented heuristic; machine compatibility; operational feasibility; mixed-integer linear programming; workstation assignment

Check for updates

Academic Editor: Nik Bessis

Received: 4 October 2026
Revised: 21 October 2026
Accepted: 26 October 2026
Published: 3 November 2026

**Copyright:** © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

## 1. Introduction

Garment manufacturing is a labor-intensive industry in which production performance depends on the effective coordination of operations, workstations, and manufacturing resources. Assembly line balancing (ALB) plays a central role in organizing these elements by distributing operations across workstations to improve resource utilization, reduce idle time, and meet production requirements. In garment assembly lines, balancing decisions are particularly challenging because operations are sub-

ject to precedence relationships, heterogeneous processing times, and specific machine requirements [1–3]. Previous studies have demonstrated the potential of line-balancing techniques and simulation-based approaches to improve productivity and operational efficiency in garment manufacturing [2–4].

The classical assembly line balancing problem primarily concerns the assignment of precedence-dependent operations to workstations under a prescribed cycle time, generally with the objective of minimizing workstation count or improving workload distribution. However, numerically balanced configurations are not necessarily technologically feasible when operations require different manufacturing equipment. Generalized assembly line balancing formulations have consequently incorporated additional constraints involving machine requirements, resource availability, workstation characteristics, and other operational restrictions [5,6]. The consideration of machine compatibility is therefore an established research direction. Nevertheless, the manner in which these constraints are incorporated into constructive task-assignment procedures remains an important methodological consideration.

This distinction is particularly relevant in garment manufacturing, where successive operations may require lockstitch, overlock, hemming, buttonhole, or finishing technologies. Bongomin et al. [2] investigated the application of the Ranked Positional Weight (RPW) method to garment assembly under different resource conditions, demonstrating that numerically favorable workstation configurations may lose their practical relevance when equipment restrictions are considered. These findings illustrate that workstation count and line efficiency alone are insufficient to establish the operational feasibility of a balancing solution. Technological compatibility must therefore be considered during assignment or explicitly verified before a configuration can be considered implementable.

More comprehensive approaches have been developed to address the complexity of garment production systems. Bongomin et al. [3] investigated a complex garment assembly line through simulation-based optimization, incorporating stochastic processing times and resource-related constraints. Metaheuristic approaches have also been applied to apparel line balancing to explore alternative workstation configurations. These methods are particularly valuable when production variability, resource interactions, and complex optimization objectives must be represented. However, depending on the modeling objectives, they may require additional operational data, simulation calibration, or algorithmic parameterization. Deterministic constructive heuristics address a complementary requirement by providing explicit and reproducible assignment decisions using a limited set of operational parameters. Their relevance is therefore not based on replacing simulation or advanced optimization, but on offering a transparent means of constructing and evaluating workstation configurations.

Classical constructive heuristics, including RPW, Shortest Processing Time (SPT), and Largest Candidate Rule (LCR), rely on different task-selection priorities. Although their conventional formulations primarily consider precedence relationships and processing times, machine compatibility can be incorporated through additional feasibility checks. Consequently, the scientific contribution of a new constructive procedure cannot be established solely by introducing machine-compatibility constraints. A more meaningful distinction concerns how operations are selected, how technological feasibility is maintained throughout workstation construction, and how the resulting configurations perform against existing heuristics operating under equivalent constraints.

Within this context, the present study proposes a Feasibility-Oriented Heuristic (FOH) for garment assembly line balancing. The methodological distinction of FOH lies in combining a predetermined topological task sequence with incremental workstation-capacity and machine-compatibility verification. Rather than ranking operations according to processing time or positional weight, FOH follows a deterministic precedence-consistent ordering and

selects the first operation satisfying the assignment conditions. For each workstation, the set of admissible machine technologies is initialized and progressively updated through intersection with the admissible machine set of each accepted operation. An assignment is permitted only when the resulting intersection remains nonempty and the available cycle-time capacity is respected.

This construction mechanism establishes a direct relationship between task selection and technological feasibility. By maintaining a nonempty common admissible machine set after each accepted assignment, FOH prevents incompatible task combinations during workstation formation rather than requiring their subsequent detection and repair. Its predefined topological sequence also provides an explicit and reproducible order of assignment decisions. Under the stated assumptions of an acyclic precedence graph, individually feasible task durations, nonempty machine-admissibility sets, and sufficient equipment availability, the procedure preserves the modeled feasibility constraints and terminates after assigning all operations. These properties are formally established in the methodological section.

The significance of FOH should nevertheless be distinguished from the feasibility checks themselves, which can also be incorporated into established priority-based heuristics. FOH offers a specific deterministic construction strategy in which task ordering is separated from technological eligibility, enabling the effects of these two elements to be examined explicitly. Its value lies in providing a formally specified, interpretable, and reproducible feasibility-oriented procedure rather than claiming that compatibility enforcement is exclusive to the proposed method. The approach is not designed to guarantee globally optimal workstation counts; its numerical performance remains dependent on the precedence structure, processing-time distribution, and machine requirements of the considered instance.

To investigate the proposed mechanism, FOH is evaluated using a 17-operation garment assembly benchmark and compared with three conventional heuristics: RPW, SPT, and LCR. To establish a fair comparison, machine-compatibility-aware variants of these methods, denoted RPW-MC, SPT-MC, and LCR-MC, are also considered. These variants retain their original selection priorities while enforcing the same technological feasibility conditions as FOH. An exact Mixed-Integer Linear Programming (MILP) formulation is employed to determine the minimum feasible workstation count and provide an optimal reference for evaluating the constructive solutions.

The experimental assessment considers workstation count, idle time, line efficiency, workload smoothness, and precedence and machine-compatibility violations. Nine deterministic scenarios combining alternative cycle-time limits and processing-time variations are examined to investigate the sensitivity of the resulting assignments. This comparative design distinguishes the effect of machine-compatibility enforcement from that of task-selection priorities and provides an objective basis for evaluating the proposed heuristic [5–10].

The study contributes to garment assembly line balancing through the formal specification of a deterministic feasibility-oriented construction procedure, the mathematical justification of its feasibility-preservation property under defined assumptions, and a comparative evaluation against conventional heuristics, compatibility-aware alternatives, and an exact optimization reference. These elements provide a structured framework for analyzing the relationship between technological feasibility and numerical balancing performance.

The experimental investigation is restricted to a single 17-operation benchmark and its deterministic variations. Although this setting allows detailed verification of assignment decisions and comparison with exact solutions, it does not establish the computational

scalability or general effectiveness of FOH on larger industrial routing matrices. Additional evaluations involving independent garment styles, larger precedence networks, and more heterogeneous machine requirements are therefore necessary. Further extensions may also incorporate machine-quantity restrictions, stochastic processing times, and operator-specific characteristics to improve the representation of industrial operating conditions.

## 2. Literature Review

Assembly line balancing has been extensively investigated through heuristic, exact, metaheuristic, and simulation-based approaches. Classical formulations primarily address the assignment of precedence-related tasks to workstations under a prescribed cycle time, whereas generalized formulations extend this structure by incorporating technological, resource, spatial, workforce, and other assignment restrictions [5–10]. In garment manufacturing, these extensions are particularly relevant because sewing operations frequently require heterogeneous machine technologies, and numerically attractive task groupings may not necessarily correspond to implementable workstation configurations [2,5].

Accordingly, the present review focuses on four areas directly related to the scope of this study: classical assembly line balancing methods, resource- and machine-constrained formulations, garment-specific balancing approaches, and the methodological gap addressed by the proposed Feasibility-Oriented Heuristic (FOH).

### *2.1. Classical Assembly Line Balancing and Benchmark Heuristics*

Assembly line balancing concerns the allocation of interdependent operations to workstations while satisfying precedence relationships and cycle-time restrictions. Depending on the production objectives, the problem may involve minimizing the number of workstations, reducing idle capacity, or improving the distribution of workload across the line. These objectives have motivated extensive research into exact optimization procedures and heuristic solution methods [5,9].

Exact approaches, including integer programming and branch-and-bound algorithms, provide a rigorous framework for identifying optimal solutions. However, tHoHheir computational requirements can increase considerably with the number of operations and the complexity of the constraints. Constructive heuristics offer a less computationally demanding alternative by assigning tasks according to predefined priority rules. Scholl and Becker [9] reviewed the principal exact and heuristic approaches developed for the simple assembly line balancing problem, highlighting the range of methods available for addressing different problem configurations.

Among conventional constructive heuristics, Ranked Positional Weight (RPW), Shortest Processing Time (SPT), and Largest Candidate Rule (LCR) are particularly relevant as benchmark methods. RPW assigns priority according to the processing time of each operation and its successors, thereby accounting for the downstream consequences of task selection. SPT favors eligible operations with shorter processing times, whereas LCR gives preference to longer operations that can be accommodated within the remaining workstation capacity. Their relatively simple decision rules facilitate implementation and allow assignment sequences to be reproduced when eligibility conditions and tie-breaking procedures are consistently specified [2,9].

Despite these advantages, the conventional formulations of these heuristics primarily address processing times and precedence relationships. Machine compatibility is not inherently incorporated into their priority rules. Consequently, operations may be grouped within the same workstation even when they require incompatible equipment. Although such assignments may produce favorable numerical balancing indicators, their

practical implementation may be impossible under the technological conditions of the production line.

This limitation does not imply that classical priority-based heuristics cannot accommodate machine restrictions. Rather, it highlights the importance of distinguishing the task-selection rule from the feasibility conditions imposed during workstation construction. Conventional heuristics can be extended to account for machine requirements, but these additional checks may influence the resulting assignment sequence and workstation configuration.

The industrial significance of resource restrictions was illustrated by Bongomin et al. [2], who investigated RPW-based balancing in garment manufacturing under different resource conditions. Their findings indicated that solutions obtained without resource restrictions could lose much of their numerical advantage when the equipment requirements of individual operations were considered.

Classical constructive heuristics therefore remain useful reference methods, particularly for examining the influence of priority rules on workload allocation. Nevertheless, their evaluation in industrial settings requires explicit consideration of technological feasibility. This observation provides a basis for examining generalized assembly line balancing approaches in which equipment and resource restrictions form part of the assignment problem [6,9].

### *2.2. Resource-, Equipment-, and Machine-Constrained Assembly Line Balancing*

The assumptions underlying the Simple Assembly Line Balancing Problem are not always representative of industrial production systems. In practice, the allocation of operations depends not only on processing times and precedence relationships but also on equipment characteristics, resource availability, and other workstation-specific requirements. Generalized assembly line balancing formulations have been developed to represent these additional conditions within the balancing problem [5].

Resource-constrained assembly line balancing has received considerable attention in the optimization literature. Moon et al. [10] investigated line balancing under resource restrictions, demonstrating the relevance of incorporating manufacturing resources into workstation assignment decisions. Pereira [8] subsequently addressed a cost-oriented, resource-constrained multi-model assembly line balancing problem, illustrating the relationship between resource requirements and the economic consequences of workstation configurations.

Other contributions have considered a broader range of industrial constraints. Kara et al. [11] developed an integrated model incorporating equipment requirements, worker skills, and ergonomic considerations. Their study demonstrated that the feasibility of workstation assignments may depend on several interacting factors that cannot be adequately represented through conventional balancing criteria alone.

The integration of assignment restrictions has also been investigated through both heuristic and exact solution procedures. Pearce [7] examined generalized assembly line balancing in an industrial context, considering constructive and improvement heuristics alongside mathematical optimization. Pearce et al. [6] further developed an integer-programming formulation for a general assembly line balancing problem involving parallel workers and additional assignment restrictions. Together, these contributions establish that incorporating technological and resource constraints is an existing and important direction in assembly line balancing research.

An essential consideration in machine-constrained balancing is how technological compatibility is represented. When an operation can be performed using one or more machine types, its requirements may be described by a set of admissible technologies. Under

the assumption that each workstation operates with one principal machine technology, a combination of operations is feasible only when their admissible machine sets have at least one machine type in common. This formulation provides a direct means of evaluating whether a proposed task grouping satisfies the specified technological requirements.

However, the representation of machine compatibility and the procedure used to construct workstation assignments are distinct methodological considerations. A mathematical formulation can define the conditions that characterize a feasible solution without prescribing the order in which operations should be selected. Similarly, constructive heuristics may apply identical feasibility conditions while generating different workstation configurations because of their task-priority rules.

This distinction is particularly important when assessing the contribution of a new constructive procedure. Its performance should be examined not only in relation to methods that disregard technological restrictions but also against alternative assignment rules operating within the same feasible solution space.

The relevance of these considerations becomes especially apparent in garment manufacturing, where production operations involve heterogeneous equipment and closely connected processing sequences. The following section examines how the particular characteristics of garment assembly lines have been addressed through heuristic, simulation-based, and optimization approaches.

### *2.3. Garment Assembly Line Balancing and Operational Constraints*

Garment assembly lines are characterized by sequences of interdependent operations requiring different processing times and manufacturing technologies. Sewing, overlocking, hemming, buttonhole making, and finishing activities may rely on distinct machine configurations, which restricts the combinations of operations that can be assigned to individual workstations. These characteristics introduce technological considerations that extend beyond the conventional objectives of minimizing idle time or reducing the number of workstations [1–3].

Chen et al. [1] investigated assembly line balancing in the garment industry, contributing to the development of systematic approaches for organizing sewing operations. Subsequent studies have examined the effects of resource restrictions, production variability, and alternative optimization methods on the performance of garment manufacturing lines.

Bongomin et al. [2] investigated the effect of resource constraints on a trouser assembly line using the RPW method. Their results showed that a configuration obtained without resource restrictions required 27 workstations and achieved a line efficiency of 80.56%. When resource requirements were considered, the number of workstations increased to 39, while line efficiency decreased to 55.77%. This difference illustrates the extent to which technological restrictions can influence balancing outcomes. It also demonstrates why workstation count and calculated line efficiency should not be interpreted independently of the conditions under which the assignments are obtained.

Simulation has been used to examine garment production systems in greater operational detail. Sime et al. [12] investigated the feasibility of simulations techniques for apparel line balancing, highlighting their potential for evaluating alternative production arrangements. Teshome et al. [4] employed Arena simulation to assess garment assembly line performance and reported improvements in production output, labor productivity, and line efficiency following the evaluation of alternative configurations.

Bongomin et al. [3] extended this research through simulation-based optimization of a complex garment assembly line comprising 69 workstations. Their model incorporated stochastic processing times, precedence relationships, resource quantities, and re-

source types. By considering these interacting factors, the study demonstrated the value of simulation-based optimization for analyzing production systems in which operational variability and resource requirements affect line performance.

Metaheuristic methods have provided another direction for addressing garment assembly line balancing. Xu et al. [13] investigated the application of adaptive ant colony optimization to apparel assembly lines, illustrating how search-based optimization can be employed to explore alternative task assignments. In a related area of garment production management, Mouloud et al. [14] examined order scheduling in the Moroccan fast-fashion industry through a clustering-based approach, emphasizing the broader importance of production heterogeneity in manufacturing decisions.

The reviewed studies demonstrate that garment production problems can be addressed at different levels of operational complexity. Simulation-based approaches are particularly useful when processing-time variability, resource interactions, and production disturbances need to be represented explicitly. Metaheuristic methods offer opportunities to explore larger and more complex solution spaces, although their implementation may require algorithmic parameterization and additional computational effort.

Deterministic constructive heuristics address a different practical requirement. When reliable production data are limited to processing times, precedence relationships, and machine requirements, a clearly defined assignment procedure may provide a useful basis for constructing and evaluating workstation configurations. The resulting decisions can be traced to explicit eligibility and selection rules, facilitating their interpretation and reproduction.

These characteristics do not establish the superiority of deterministic heuristics over simulation or metaheuristic optimization. Rather, they suggest that different approaches may be appropriate depending on the available data, the complexity of the production environment, and the objectives of the analysis. A deterministic method may be sufficient for generating feasible initial configurations, whereas simulation and advanced optimization can support more detailed investigations of system behavior and solution quality.

A central consideration emerging from the literature is therefore the need to distinguish the numerical performance of a balancing procedure from the technological feasibility of its assignments. Equally important is the need to assess whether improvements in feasibility arise from the underlying task-selection mechanism or from the additional constraints imposed during workstation construction. These considerations inform the methodological positioning of the present study.

### *2.4. Research Gap and Positioning of the Present Study*

The literature confirms that machine and resource restrictions are established considerations in assembly line balancing. Generalized formulations have incorporated equipment requirements, workforce characteristics, and additional assignment constraints, while garment-specific studies have examined resource-constrained heuristics, simulation-based optimization, and metaheuristic procedures [2–13]. Accordingly, the contribution of the present research cannot be attributed to the introduction of machine compatibility as a previously unexplored balancing constraint.

The relevant methodological issue concerns how technological feasibility is incorporated into the construction of workstation assignments and how the resulting solutions are evaluated. Conventional priority-based heuristics offer clearly defined task-selection rules, but machine restrictions must be introduced explicitly when they are not part of the original assignment procedure. Such modifications can influence both the number of workstations and the distribution of operations across the line.

Although resource-aware heuristics and more advanced optimization methods have already been investigated, the literature considered in this study provides a basis for further examining deterministic, garment-oriented constructive procedures whose feasibility conditions and assignment decisions are fully specified and reproducible. Of particular interest is the comparative assessment of such procedures under common precedence, cycle-time, and machine-compatibility requirements, rather than relying exclusively on comparisons with conventional heuristics operating under less restrictive assumptions.

To investigate this issue, the present study proposes a Feasibility-Oriented Heuristic (FOH) for garment assembly line balancing. The method constructs workstations sequentially using a deterministic topological ordering of operations. At each assignment step, an operation is considered only when its precedence requirements have been satisfied, its processing time can be accommodated within the remaining workstation capacity, and its admissible machine set is compatible with the current workstation configuration.

The distinction between FOH and conventional priority-based heuristics lies in the combination of its task-selection sequence and the systematic enforcement of feasibility conditions throughout workstation construction. However, enforcing technological compatibility is not exclusive to FOH, since comparable checks can also be incorporated into existing heuristics. Its contribution must therefore be assessed through the resulting assignment behavior and solution quality rather than through the presence of machine constraints alone.

For this reason, the evaluation distinguishes between numerical balancing performance and technological feasibility. Conventional implementations of RPW, SPT, and LCR provide a reference for examining the consequences of assigning tasks without machine-compatibility enforcement. Compatibility-aware variants of these methods offer a complementary basis for comparing alternative constructive rules under equivalent technological conditions. An exact Mixed-Integer Linear Programming (MILP) formulation further provides an optimal workstation-count reference for instances that can be solved to proven optimality.

The practical relevance of FOH lies in providing an explicit and interpretable procedure for generating workstation configurations that satisfy the modeled machine requirements. Its deterministic structure allows assignment decisions to be reproduced from the same operational data without the algorithmic parameterization associated with more elaborate search procedures. Nevertheless, the method does not guarantee globally optimal solutions, and its performance may depend on the precedence structure, processing-time distribution, and technological characteristics of the production line.

The present investigation is restricted to deterministic processing times, precedence constraints, cycle-time capacity, and task–machine compatibility. Machine-quantity limitations, operator-specific capabilities, learning effects, fatigue, and stochastic production disturbances are not explicitly represented. Furthermore, the relatively small benchmark used in the experimental evaluation limits the extent to which conclusions can be generalized to larger industrial instances.

Within this defined scope, the study contributes a formally specified constructive approach and a comparative framework for examining the relationship between assignment decisions, technological feasibility, and conventional balancing indicators. Rather than replacing established optimization methods, FOH is positioned as a complementary procedure whose practical value and limitations can be examined against classical heuristics and an exact feasibility-constrained reference.

## 3. Materials and Methods

### *3.1. Problem Definition and Benchmark Dataset*

This study considers a garment assembly line balancing problem in which precedence-dependent operations must be allocated to ordered workstations under cycle-time and machine-compatibility constraints. The formulation extends conventional workload-based assignment by requiring all operations grouped within a workstation to share at least one admissible machine technology.

Let $I = \{1,2,...,n\}$ denote the set of operations, where n is the total number of tasks. Each operation $i \in I$ is characterized by a deterministic processing time $t_i > 0$, a set of immediate predecessors $\mathrm{Pred}(i)$, and a nonempty set $M_i$ of admissible machine types. The complete set of machine types is denoted by $\mathcal{M} = \bigcup_{i \in I} M_i$.

The required production pace is determined by the takt time:

$$\boldsymbol{TT} = \frac{\boldsymbol{T_{available}}}{\boldsymbol{D}} \tag{1}$$

where $T_{available}$ is the net available production time and D is the required production quantity over the same planning period. When the production rate is determined by demand, the nominal workstation cycle time is set to:

$$\boldsymbol{C_0} = \boldsymbol{TT} \tag{2}$$

In this study, $C_0 = 2.00$ min is specified as the nominal experimental condition, rather than calculated from independently observed demand data.

The total work content and theoretical minimum number of workstations are defined as:

$$\boldsymbol{W} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{t_i} \tag{3}$$

$$\boldsymbol{N_{min}} = \left\lceil \frac{\boldsymbol{W}}{\boldsymbol{C}} \right\rceil \tag{4}$$

where C is the prescribed cycle time and $\lceil \cdot \rceil$ denotes the ceiling function. This lower bound does not account for precedence or machine-compatibility restrictions.

The investigated benchmark comprises 17 garment assembly operations with a total work content of 20.00 min. Under the nominal cycle time:

$$\boldsymbol{N_{min}} = \left\lceil \frac{\boldsymbol{20.00}}{\boldsymbol{2.00}} \right\rceil \boldsymbol{=10} \tag{5}$$

This value represents a workload-based lower bound rather than the minimum feasible workstation count.

The processing times, precedence relationships, and admissible machine types are presented in Table 1.

Each operation must be assigned exactly once, without task splitting, while respecting precedence relationships and workstation capacity. Operations sharing a workstation are executed in an order consistent with the precedence graph.

Machine compatibility is modeled through the admissible machine sets. For a nonempty workstation k containing the operations $S_k$, technological feasibility requires:

$$\boldsymbol{M}(\boldsymbol{S_k}) = \bigcap_{\boldsymbol{i \in S_k}} \boldsymbol{M_i} \neq \emptyset \tag{6}$$

where $M(S_k)$ denotes the set of machine technologies common to all operations assigned to workstation k, and $M_i$ represents the admissible machine set of operation i. A nonempty intersection ensures that all operations assigned to the workstation can be performed using at least one common machine technology.

**Table 1.** Operational dataset of the 17-task garment assembly benchmark instance.

| Task | Operation | Processing Time (min) | Admissible Machine Type (s) | Immediate Predecessor (s) |
|---|---|---|---|---|
| **OP1** | Fusing collar and collar stand | 1.20 | Fusing press | — |
| **OP2** | Preparing chest pocket | 0.80 | Single-needle lockstitch | — |
| **OP3** | Attaching chest pocket to front panel | 0.70 | Single-needle lockstitch | OP2 |
| **OP4** | Preparing button placket | 1.10 | Single-needle lockstitch | — |
| **OP5** | Assembling back yoke | 1.00 | Single-needle lockstitch | — |
| **OP6** | Joining shoulders | 1.00 | Single-needle lockstitch | OP3, OP4, OP5 |
| **OP7** | Preparing collar | 1.40 | Single-needle lockstitch | OP1 |
| **OP8** | Attaching collar to garment body | 1.20 | Single-needle lockstitch | OP6, OP7 |
| **OP9** | Preparing sleeves | 1.40 | Single-needle lockstitch | — |
| **OP10** | Setting sleeves into armholes | 1.30 | Overlock or lockstitch | OP6, OP9 |
| **OP11** | Closing left side seam and under-sleeve seam | 1.25 | Overlock | OP8, OP10 |
| **OP12** | Closing right side seam and under-sleeve seam | 1.25 | Overlock | OP8, OP10 |
| **OP13** | Topstitching and collar placket finishing | 1.10 | Single-needle lockstitch | OP8 |
| **OP14** | Bottom hemming | 0.90 | Hemming or coverstitch | OP11, OP12 |
| **OP15** | Making buttonholes | 1.30 | Buttonhole machine | OP13, OP14 |
| **OP16** | Sewing buttons | 1.40 | Button-sewing machine | OP15 |
| **OP17** | Final pressing, inspection, and folding | 1.70 | Ironing/finishing station | OP16 |
| **Total** | Total work content | 20.00 | — | — |

The model assumes one principal machine technology per workstation. Operations requiring different attachments or guides may share a workstation when they remain executable on the same underlying machine technology. However, attachment-change times and configurations involving several distinct primary machines are not explicitly represented. The benchmark also assumes sufficient machine availability and does not impose equipment-quantity restrictions. OP10 is represented as compatible with two machine categories, whereas the hemming/coverstitch designation for OP14 is treated as a single technological category in the computational model.

Processing times are deterministic, and operator-related characteristics, including proficiency, fatigue, and learning effects, are excluded. Feasibility is therefore assessed with respect to the modeled precedence, cycle-time, and technological constraints rather than the full range of industrial operating conditions.

### *3.2. Classical Benchmark Heuristics and Compatibility-Aware Variants*

Three conventional constructive heuristics are considered: Ranked Positional Weight (RPW), Shortest Processing Time (SPT), and Largest Candidate Rule (LCR). These methods provide reproducible benchmarks based on different task-selection priorities.

RPW selects the eligible operation with the highest positional weight, calculated as:

$$\boldsymbol{PW_i = t_i + \sum_{j \in \mathbf{Succ}^*(i)} t_j} \tag{7}$$

where $\mathrm{Succ}^*(i)$ denotes the set of direct and indirect successors of operation i, each counted once. SPT prioritizes the eligible operation with the shortest processing time, whereas LCR selects the longest eligible operation that fits the remaining workstation capacity. Ties are resolved by ascending numerical task identifier.

Two types of implementation are evaluated. In their conventional forms, RPW, SPT, and LCR enforce precedence and cycle-time constraints during assignment, and machine compatibility is verified afterward. These implementations provide a reference for identifying numerical improvements that may depend on technologically infeasible task combinations.

To establish a comparison under equivalent constraints, three compatibility-aware variants are introduced: RPW-MC, SPT-MC, and LCR-MC. Each retains its original priority rule but verifies machine compatibility before assigning an operation. The admissible machine-set intersection is updated after every accepted assignment, using the condition defined in Equation (6).

All constructive methods use the same precedence-eligibility definition, workstation-capacity restrictions, station-opening and closure rules, and deterministic tie-breaking conventions. The conventional and compatibility-aware implementations of a given heuristic differ only in whether machine compatibility is enforced during assignment.

This distinction separates the effect of machine-feasibility enforcement from that of task-selection priority. Accordingly, the compatibility-aware variants serve as the principal heuristic references for evaluating FOH, and the conventional methods are retained to illustrate the consequences of neglecting technological restrictions.

### *3.3. Proposed Feasibility-Oriented Heuristic*

The proposed Feasibility-Oriented Heuristic (FOH) is a deterministic constructive procedure that integrates precedence eligibility, cycle-time capacity, and machine compatibility into workstation formation. Unlike compatibility-aware priority heuristics, FOH selects operations according to a fixed topological sequence rather than processing-time or positional-weight priorities. Its construction mechanism combines this predefined ordering with the continuous verification of technological feasibility throughout task assignment.

The precedence structure is represented by a directed acyclic graph:

$$\boldsymbol{G = (I, P)} \tag{8}$$

where I denotes the set of operations and P the set of precedence arcs. A deterministic topological ordering $\pi$ is generated by selecting the smallest task identifier whenever several operations are simultaneously available during the sorting procedure.

Let U denote the set of unassigned operations and A the set of already assigned operations. The precedence-eligible operations are defined as:

$$\boldsymbol{E(A) = \{i \in U: \mathbf{Pred}(i) \subseteq A\}} \tag{9}$$

where $\mathrm{Pred}(i)$ represents the set of immediate predecessors of operation i.

For workstation k, let $S_k$ represent its assigned operations, $L_k$ its accumulated processing load, and $M_k$ the set of machine types that remain admissible.

A newly opened workstation is initialized with

$$\boldsymbol{S_k = \emptyset, L_k = 0 \text{ And } M_k = \mathcal{M}.}$$

where $\mathcal{M}$ denotes the complete set of admissible machine technologies.

An operation can be assigned only if it satisfies precedence eligibility, remaining workstation capacity, and machine compatibility. The feasible candidate set is therefore:

$$\boldsymbol{F_k = \{i \in E(A): L_k + t_i \leq C, M_k \cap M_i \neq \emptyset\}} \quad (10)$$

where $t_i$ is the processing time of operation i, C is the prescribed cycle time, and $M_i$ denotes the admissible machine set of operation i.

FOH selects the first feasible operation $i \in F_k$ according to the predetermined topological sequence $\pi$.

Following the assignment of operation i, the workstation state is updated as follows:

$$\begin{gathered} \boldsymbol{S_k \leftarrow S_k \cup \{i\}} \\ \boldsymbol{L_k \leftarrow L_k + t_i} \\ \boldsymbol{M_k \leftarrow M_k \cap M_i} \\ \boldsymbol{A \leftarrow A \cup \{i\}} \\ \boldsymbol{U \leftarrow U \setminus \{i\}} \end{gathered} \quad (11)$$

The procedure continues until no additional feasible operation can be assigned to the current workstation. A new workstation is then opened if unassigned operations remain. Algorithm 1 summarizes the complete constructive procedure. The corresponding workstation-construction workflow is illustrated in Figure 1.

- **Feasibility-Preservation Property:**

The constructive nature of FOH allows its feasibility-preservation property to be established under the assumptions of the proposed model.

**Proposition 1.** *Feasibility preservation and finite termination of FOH.*

Consider a finite set of operations I, an acyclic precedence graph G = (I,P), and a positive cycle time C.

Assume that each operation i has a strictly positive processing time satisfying:

$$\boldsymbol{0 < t_i \leq C, \forall\ i \in I.}$$

and a nonempty admissible machine set:

$$\boldsymbol{M_i \neq \emptyset, \forall\ i \in I.}$$

Under the assumption of sufficient machine availability and without additional equipment-quantity restrictions, FOH terminates after a finite number of assignments and produces a workstation configuration satisfying the modeled precedence, cycle-time, and machine-compatibility constraints.

**Algorithm 1. Feasibility-Oriented Heuristic (FOH)**

**Input:** Task set I; processing times $t_i$; precedence relations P; admissible machine sets $M_i$; global machine set $\mathcal{M}$; cycle time C.
**Output:** Ordered workstation assignments $S_1, S_2, \ldots, S_m$ and workstation count m.

1: Construct the precedence graph G = (I, P).
2: Perform a deterministic topological sort of G using ascending task identifiers to resolve ties.
3: **if** a complete topological ordering cannot be obtained then
  4: Report an infeasible precedence structure.
  5: Stop.
6: **end if**
7: Store the resulting topological sequence $\pi$.
8: U ← I. (Set of unassigned tasks.)
9: A ← ∅. (Set of assigned tasks.)
10: k ← 1. (Workstation index.)
11: **while** U ≠ ∅ do
  12: Open workstation k.
  13: Sk ← ∅.
  14: Lk ← 0.
  15: Mk ← $\mathcal{M}$.
  16: **while** true do
    17: E ← {i ∈ U : Pred(i) ⊆ A}.
    18: Fk ← {i ∈ E : Lk + ti ≤ C and Mk ∩ Mi ≠ ∅}.
    19: **if** Fk = ∅ then
      20: **if** Sk = ∅ then
        21: Report an infeasible instance.
        22: Stop.
      23: **else**
        24: Close workstation k.
        25: **break.**
      26: **end if**
    27: **end if**
    28: Select the first task i ∈ Fk according to the topological sequence $\pi$.
    29: Sk ← Sk ∪ {i}.
    30: Lk ← Lk + ti.
    31: Mk ← Mk ∩ Mi.
    32: A ← A ∪ {i}.
    33: U ← U ∖ {i}.
    34: **if** U = ∅ then
      35: **break.**
    36: **end if**
  37: **end while**
  38: **if** U ≠ ∅ then
    39: k ← k + 1.
  40: **end if**
41: **end while**
42: m ← k; return S1, S2, ..., Sm and m.

**Proof.** Precedence feasibility is preserved because an operation is assigned only when all its predecessors belong to the set of already assigned operations. Consequently, each predecessor is allocated either to an earlier workstation or to the same workstation before its successor. □

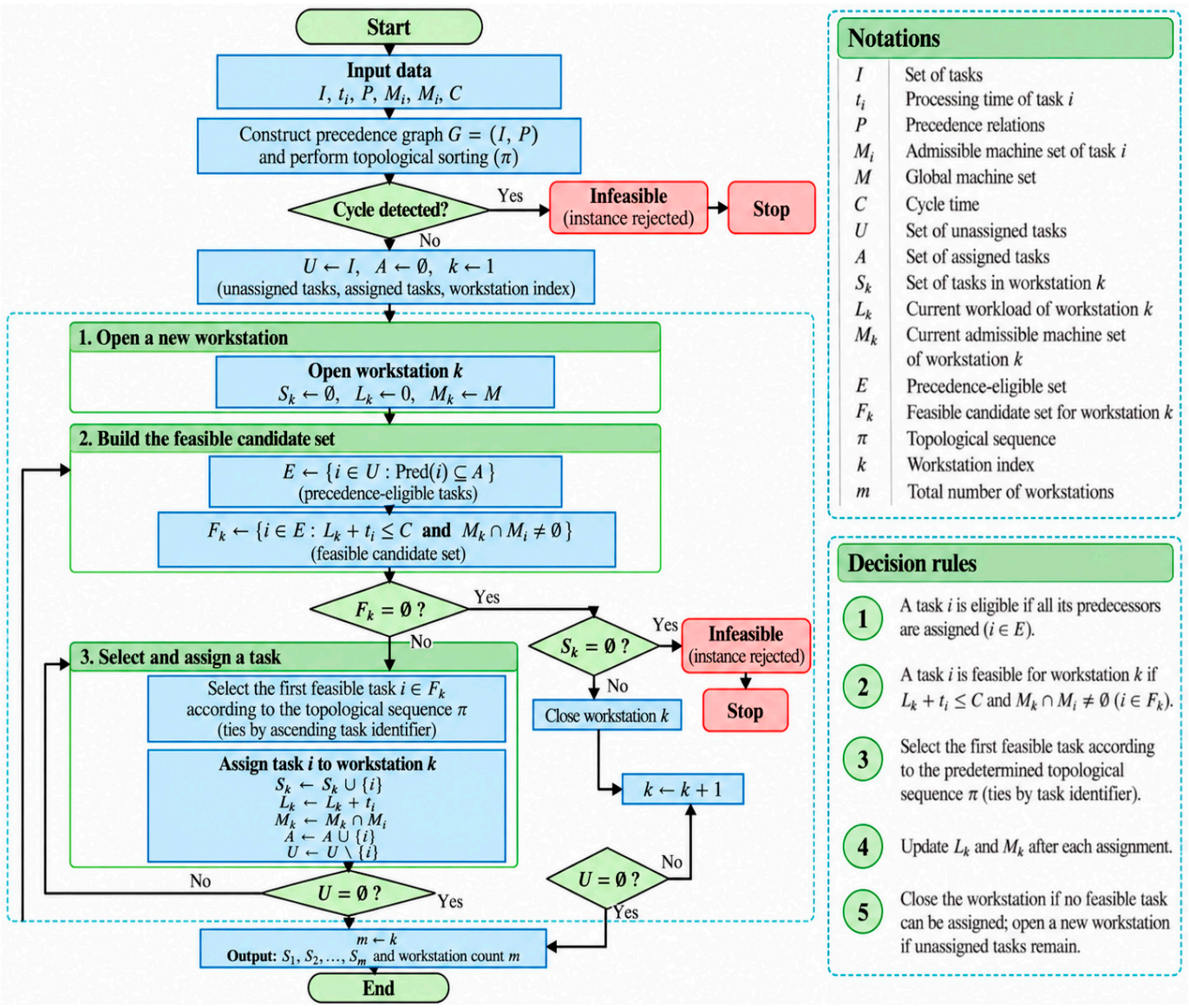



**Figure 1.** Flowchart of the Feasibility-Oriented Heuristic (FOH).

Cycle-time feasibility follows from the assignment condition:

$$L_k + t_i \leq C.$$

Since the workstation load is initialized at zero and updated only after satisfying this condition, its accumulated processing time never exceeds the prescribed cycle time.

Machine compatibility is preserved through the iterative update:

$$M_k \leftarrow M_k \cap M_i.$$

An operation is accepted only if the intersection is nonempty. Therefore, every constructed workstation retains at least one common admissible machine type for all its assigned operations.

Finally, because the precedence graph is acyclic, the subgraph induced by the remaining unassigned operations contains at least one operation whose predecessors have already been assigned. Such an operation is precedence-eligible.

If no additional operation can be accommodated in the current workstation, a new workstation is opened. Since each operation satisfies the individual cycle-time requirement

and has a nonempty admissible machine set, at least one precedence-eligible operation can be assigned to an empty workstation.

Each successful assignment removes one operation from the finite set of unassigned operations. Therefore, FOH terminates after exactly n successful task assignments and opens at most n workstations.

This establishes the stated feasibility-preservation and finite-termination properties.

- **Application to the Nominal Benchmark**

For the nominal benchmark, OP1 is allocated to WS1 with a workload of 1.20 min. No other precedence-eligible operation shares its fusing-press requirement.

WS2 then receives OP2 and OP3, both requiring a lockstitch machine, resulting in a total workload of:

$$L_2 = t_2 + t_3$$

$$L_2 = 0.08 + 0.70 = 1.50 \text{ min}$$

The same assignment mechanism is applied until all 17 operations have been allocated.

- **Methodological Contribution and Positioning of FOH**

The methodological contribution of FOH lies in the explicit integration of deterministic topological task ordering with incremental workstation-capacity and machine-compatibility verification. Its construction mechanism provides a fully specified and reproducible sequence of assignment decisions, and the feasibility-preservation property establishes the validity of the resulting configurations under the stated assumptions.

Although compatibility-aware classical heuristics can satisfy the same feasibility requirements, they employ different task-selection priorities and may consequently produce different workstation configurations. FOH is therefore distinguished by its deterministic topological construction strategy rather than by the introduction of machine compatibility as a new constraint.

The proposed procedure does not guarantee workstation-count optimality, and its numerical performance is evaluated against alternative compatibility-aware heuristics and the exact MILP reference in Section 4.

### *3.4. Exact MILP Reference and Performance Measures*

#### 3.4.1. Mixed-Integer Linear Programming Formulation

An exact Mixed-Integer Linear Programming (MILP) model is formulated to determine the minimum feasible workstation count under the same constraints considered by FOH.

Let $K = \{1,\ldots, n\}$ denote the set of potential workstation positions. The binary variables $x_{ik}$, $y_k$ and $z_{kr}$ indicate, respectively, whether operation i is assigned to workstation k, whether workstation k is opened, and whether workstation k uses machine type r. The binary parameter $a_{ir}$ equals one when $r \in M_i$ and zero otherwise.

The objective function is:

$$\min \sum_{k \in K} y_k \tag{12}$$

subject to:

**Unique task assignment**

$$\sum_{k \in K} x_{ik} = 1, \quad \forall i \in I \tag{13}$$

Equation (13) ensures that each operation is assigned to exactly one workstation.

**Workstation cycle-time capacity**

$$\sum_{i \in I} t_i x_{ik} \leq C y_k, \quad \forall k \in K \tag{14}$$

Equation (14) restricts the total processing time assigned to an opened workstation to the prescribed cycle-time limit. Since all processing times are strictly positive, the constraint also prevents operations from being assigned to closed workstations.

**Precedence relationships**

$$\sum_{k \in K} k x_{ik} \leq \sum_{k \in K} k x_{jk}, \quad \forall (i,j) \in P \tag{15}$$

Equation (15) requires each predecessor to be assigned to the same workstation as, or a workstation preceding, its successor. Operations allocated to the same workstation are assumed to be executed in an order consistent with the acyclic precedence graph.

**Task–machine compatibility**

$$x_{ik} \leq \sum_{r \in \mathcal{M}} a_{ir} z_{kr}, \quad \forall i \in I,\ k \in K \tag{16}$$

Equation (16) permits an operation to be assigned to a workstation only when the workstation is configured with a machine type admissible for that operation.

**Workstation machine configuration**

$$\sum_{r \in \mathcal{M}} z_{kr} = y_k, \quad \forall k \in K \tag{17}$$

Equation (17) ensures that each opened workstation is associated with exactly one principal machine technology and that closed workstations have no assigned machine type.

Together, Equations (16) and (17) provide a binary representation of the common-machine compatibility condition established in Equation (6).

**Consecutive workstation opening**

$$y_k \geq y_{k+1}, \quad k = 1,\ldots, n-1 \tag{18}$$

Equation (18) requires opened workstations to occupy consecutive positions, eliminating certain equivalent workstation-indexing configurations without changing the optimal objective value.

**Workstation occupancy constraint**

$$y_k \leq \sum_{i \in I} x_{ik}, \quad \forall k \in K \tag{19}$$

Equation (19) prevents a workstation from being declared open when no operation is assigned to it. This strengthening constraint ensures consistency between the mathematical formulation and the computational implementation.

Finally, all decision variables are binary:

$$x_{ik},\ y_k,\ z_{kr} \in \{0,1\}$$

for all applicable indices.

The resulting MILP formulation minimizes workstation count while enforcing unique task assignment, precedence relationships, cycle-time capacity, and machine compatibility.

The exact model and the compatibility-aware constructive heuristics therefore operate within the same constrained solution space. Their distinction lies in the solution procedure: the MILP identifies a minimum-workstation solution when optimality is proven, whereas FOH and the other constructive methods generate feasible configurations according to predefined selection rules.

It should be emphasized that the MILP objective is restricted to minimizing the number of workstations. Workload smoothness is not included as either a primary or secondary optimization criterion. Consequently, different optimal MILP assignments may have identical workstation counts while exhibiting different workload distributions and Smoothness Index values.

3.4.2. Performance and Feasibility Indicators

For a configuration containing m workstations, total idle time, line efficiency, and balance delay are calculated as:

$$\boldsymbol{IT = mC - W} \tag{20}$$

$$\boldsymbol{LE = \frac{W}{mC} \times 100} \tag{21}$$

$$\boldsymbol{BD = 100 - LE} \tag{22}$$

Workload distribution is assessed using the Smoothness Index:

$$\boldsymbol{SI = \sqrt{\sum_{k=1}^{m} (L_{max} - L_k)^2}} \tag{23}$$

where m denotes the number of workstations, C the prescribed cycle time, W the total work content, $L_k$ the processing load of workstation k, and $L_{\max}$ the maximum workstation load. Lower SI values indicate a more uniform workload distribution, although comparisons involving different workstation counts should be interpreted with caution.

Machine-compatibility violations are quantified as:

$$\boldsymbol{V_M = \sum_{k=1}^{m} 1[M(S_k) = \emptyset]} \tag{24}$$

Precedence violations are calculated as:

$$\boldsymbol{V_P = \sum_{(i,j) \in P} 1[s(i) > s(j)]} \tag{25}$$

where s(i) denotes the workstation assigned to operation i, and $\mathbf{1}[\cdot]$ is the indicator function.

A solution is considered feasible when every operation is assigned exactly once, workstation capacities are respected, $V_M = 0$, $V_P = 0$, and same-station operations can be executed consistently with the precedence graph.

For a feasible heuristic solution H, the relative workstation-count optimality gap is:

$$\boldsymbol{Gap_H = \frac{N_H - N_{MILP}}{N_{MILP}} \times 100} \tag{26}$$

where $N_H$ is the heuristic workstation count and $N_{MILP}$ is the proven MILP optimum. This gap is not calculated for technologically infeasible heuristic configurations.

### *3.5. Experimental Protocol and Computational Validation*

Comparative Experimental Protocol

Eight approaches are evaluated using the same benchmark data: RPW, SPT, LCR, their compatibility-aware variants RPW-MC, SPT-MC and LCR-MC, FOH, and the exact MILP reference. The conventional heuristics are assessed for machine compatibility after assignment, whereas the four feasibility-constrained constructive procedures enforce this requirement during workstation construction.

The sensitivity analysis combines three cycle-time levels, $C \in \{1.80, 2.00, 2.20\}$ min, with three deterministic processing-time variations, $\boldsymbol{\delta} \in \{\mathbf{-5}\%, \mathbf{0}, \mathbf{+5}\%\}$,

Processing times are modified uniformly according to:

$$\boldsymbol{t_i^{(\delta)} = t_i(1 + \delta)} \tag{27}$$

where $t_i^{(\delta)}$ represents the adjusted processing time, and $\delta$ denotes the relative variation factor.

Precedence relationships and machine requirements remain unchanged, producing nine deterministic scenarios. These perturbations represent controlled changes in operation durations rather than stochastic simulations or empirically established worst-case limits.

The computational verification uses Python 3.13.5 and SciPy 1.17.0, with the MILP solved through the scipy.optimize.milp interface and HiGHS [15,16]. Presolve is enabled, the requested relative MIP-gap tolerance is set to zero, and a 90 s time limit is applied to each instance. A numerical tolerance of $10^{-9}$ min is used for floating-point capacity comparisons.

All reported assignments are checked for task uniqueness, precedence, cycle-time compliance, and machine compatibility. MILP solutions are treated as exact references only when the solver reports proven optimality. Execution times, branch-and-bound node counts, and final MIP gaps are recorded separately for each scenario.

Since workstation-count minimization is the sole MILP objective, different optimal assignments may exhibit different Smoothness Index values. The smoothness measure of the originally reported optimal assignment is therefore distinguished from the values and solver statistics obtained during independent computational verification.

Finally, the nine scenarios are different versions of one 17-operation benchmark, not nine independent industrial validation cases. The findings are consequently limited to the investigated precedence structure, processing-time conditions, and machine-admissibility relationships. Validation on larger and independent instances remains necessary to assess scalability and generalizability.

## 4. Results and Interpretations

### *4.1. Nominal Benchmark Results*

The eight solution approaches were first evaluated under the nominal operating condition defined by a cycle time of: $C = 2.00$ min and a total work content of: $W = 20.00$ min

The theoretical lower bound based only on total workload is therefore

$$N_{min} = \left\lceil \frac{20}{2} \right\rceil = 10$$

However, this lower bound does not account for precedence or machine-compatibility restrictions. The numerical and feasibility results obtained under the nominal condition are summarized in Table 2.

**Table 2.** Comparative performance under the nominal operating condition.

| Method | WS | Idle Time (min) | LE (%) | BD (%) | SI | $V_M$ | Feasible |
|---|---|---|---|---|---|---|---|
| RPW | 14 | 8.00 | 71.43 | 28.57 | 2.378 | 1 | No |
| SPT | 16 | 12.00 | 62.50 | 37.50 | 1.969 | 0 | Yes |
| LCR | 13 | 6.00 | 76.92 | 23.08 | 1.989 | 2 | No |
| RPW-MC | 15 | 10.00 | 66.67 | 33.33 | 2.414 | 0 | Yes |
| SPT-MC | 16 | 12.00 | 62.50 | 37.50 | 1.969 | 0 | Yes |
| LCR-MC | 15 | 10.00 | 66.67 | 33.33 | 2.414 | 0 | Yes |
| FOH | 15 | 10.00 | 66.67 | 33.33 | 2.763 | 0 | Yes |
| MILP | 15 | 10.00 | 66.67 | 33.33 | 2.063 * | 0 | Yes |

WS: number of workstations; LE: Line Efficiency; BD: Balance Delay; SI: Smoothness Index; and $V_M$: number of machine-compatibility violations. All configurations satisfy precedence constraints. * The reported SI of 2.063 corresponds to the originally reported optimal MILP assignment. An independent computational reconstruction produced another optimal 15-workstation assignment with an SI of 2.414. Workload smoothness was not included in the MILP objective function.

Line efficiency was calculated according to Equation (21), using the total work content, workstation count, and prescribed cycle time.

LCR yields the most favorable apparent numerical indicators, requiring only 13 workstations and achieving a line efficiency of 76.92%. However, two of its workstations violate the machine-compatibility requirement. The resulting configuration is therefore infeasible under the technological assumptions of the model.

RPW similarly generates a numerically attractive 14-workstation solution with a line efficiency of 71.43%, but one machine-incompatible workstation is detected.

SPT produces a feasible configuration without machine or precedence violations, although it requires 16 workstations and consequently presents the lowest line efficiency among the evaluated methods.

FOH generates a fully feasible 15-workstation configuration with: $V_M = 0$ and $V_P = 0$.

The exact MILP reference also requires 15 workstations. Thus, for the nominal benchmark, FOH reaches the same feasible workstation count as the exact optimum.

To ensure a comparison under equivalent technological constraints, the classical heuristics were additionally evaluated using dynamic machine-compatibility checks. RPW-MC and LCR-MC also achieved feasible 15-workstation configurations, whereas SPT-MC required 16 workstations. These findings confirm that compatibility enforcement can be incorporated into different constructive procedures, whereas their workstation-count performance depends on the task-selection strategy.

Although FOH and the original MILP solution achieve the same workstation count, their workload distributions differ. FOH produces a maximum workstation load of 2.00 min, compared with 1.80 min for the originally reported MILP solution. The corresponding Smoothness Index values are 2.763 and 2.063, respectively. As smoothness is not included in the MILP objective function, this difference reflects the particular task allocations rather than an explicitly optimized workload distribution.

### *4.2. Workstation-Level Feasibility Verification*

To identify the origin of the infeasibilities observed in the aggregate results, the complete workstation assignments were examined. The detailed workstation assignments for each method are reported in Table 3.

**Table 3.** Workstation-level assignments under the nominal condition. Each entry indicates the workstation number, assigned operations, processing load in minutes, and common admissible machine type. OP numbers refer to Table 1.

| Method | Workstation Assignments: Tasks (Load; Common Machine) |
|---|---|
| **RPW** | WS1: OP2 + OP4 (1.90; L); WS2: OP5 + OP3 (1.70; L); WS3: OP1 (1.20; FP); WS4: OP6 (1.00; L); WS5: OP7 (1.40; L); WS6: OP9 (1.40; L); WS7: OP8 (1.20; L); WS8: OP10 (1.30; L/O); WS9: OP11 (1.25; O); WS10: OP12 (1.25; O); WS11: OP13 + OP14 (2.00; ∅); WS12: OP15 (1.30; BH); WS13: OP16 (1.40; BS); WS14: OP17 (1.70; IF). |
| **SPT** | WS1: OP2 + OP3 (1.50; L); WS2: OP5 (1.00; L); WS3: OP4 (1.10; L); WS4: OP6 (1.00; L); WS5: OP1 (1.20; FP); WS6: OP7 (1.40; L); WS7: OP8 (1.20; L); WS8: OP13 (1.10; L); WS9: OP9 (1.40; L); WS10: OP10 (1.30; L/O); WS11: OP11 (1.25; O); WS12: OP12 (1.25; O); WS13: OP14 (0.90; H/C); WS14: OP15 (1.30; BH); WS15: OP16 (1.40; BS); WS16: OP17 (1.70; IF). |
| **LCR** | WS1: OP9 (1.40; L); WS2: OP1 + OP2 (2.00; ∅); WS3: OP7 (1.40; L); WS4: OP4 + OP3 (1.80; L); WS5: OP5 + OP6 (2.00; L); WS6: OP10 (1.30; L/O); WS7: OP8 (1.20; L); WS8: OP11 (1.25; O); WS9: OP12 (1.25; O); WS10: OP13 + OP14 (2.00; ∅); WS11: OP15 (1.30; BH); WS12: OP16 (1.40; BS); WS13: OP17 (1.70; IF). |

**Table 3.** *Cont.*

| Method | Workstation Assignments: Tasks (Load; Common Machine) |
|---|---|
| **FOH** | WS1: OP1 (1.20; FP); WS2: OP2 + OP3 (1.50; L); WS3: OP4 (1.10; L); WS4: OP5 + OP6 (2.00; L); WS5: OP7 (1.40; L); WS6: OP8 (1.20; L); WS7: OP9 (1.40; L); WS8: OP10 (1.30; L/O); WS9: OP11 (1.25; O); WS10: OP12 (1.25; O); WS11: OP13 (1.10; L); WS12: OP14 (0.90; H/C); WS13: OP15 (1.30; BH); WS14: OP16 (1.40; BS); WS15: OP17 (1.70; IF). |
| **RPW-MC** | WS1: OP2 + OP4 (1.90; L); WS2: OP5 + OP3 (1.70; L); WS3: OP1 (1.20; FP); WS4: OP6 (1.00; L); WS5: OP7 (1.40; L); WS6: OP9 (1.40; L); WS7: OP8 (1.20; L); WS8: OP10 (1.30; L/O); WS9: OP11 (1.25; O); WS10: OP12 (1.25; O); WS11: OP13 (1.10; L); WS12: OP14 (0.90; H/C); WS13: OP15 (1.30; BH); WS14: OP16 (1.40; BS); WS15: OP17 (1.70; IF). |
| **SPT-MC** | Identical workstation assignments to SPT under the nominal condition, requiring 16 feasible workstations. |
| **LCR-MC** | WS1: OP9 (1.40; L); WS2: OP1 (1.20; FP); WS3: OP7 (1.40; L); WS4: OP4 + OP2 (1.90; L); WS5: OP5 + OP3 (1.70; L); WS6: OP6 (1.00; L); WS7: OP10 (1.30; L/O); WS8: OP8 (1.20; L); WS9: OP11 (1.25; O); WS10: OP12 (1.25; O); WS11: OP13 (1.10; L); WS12: OP14 (0.90; H/C); WS13: OP15 (1.30; BH); WS14: OP16 (1.40; BS); WS15: OP17 (1.70; IF). |
| **MILP *** | WS1: OP1 (1.20; FP); WS2: OP2 + OP5 (1.80; L); WS3: OP3 + OP4 (1.80; L); WS4: OP6 (1.00; L); WS5: OP9 (1.40; L); WS6: OP7 (1.40; L); WS7: OP8 (1.20; L); WS8: OP10 (1.30; L/O); WS9: OP12 (1.25; O); WS10: OP13 (1.10; L); WS11: OP11 (1.25; O); WS12: OP14 (0.90; H/C); WS13: OP15 (1.30; BH); WS14: OP16 (1.40; BS); WS15: OP17 (1.70; IF). |

* The MILP row represents the originally reported optimal assignment, consisting of 15 workstations with a Smoothness Index (SI) of 2.063. An independent computational reconstruction also yielded an optimal 15-workstation assignment, but with an SI of 2.414. Workload smoothness was not included in the MILP objective function.

The following machine abbreviations are used: FP = fusing press, L = lockstitch, O = overlock, H/C = hemming or coverstitch, BH = buttonhole, BS = button sewing, and IF = ironing/finishing. The symbol $\emptyset$ indicates that the tasks grouped in the workstation have no common admissible machine type.

The RPW infeasibility occurs at WS11, where OP13 and OP14 are grouped because their combined processing time exactly equals the cycle time:

$$t_{13} + t_{14} = 1.10 + 0.90 = 2.00 \text{ min}.$$

However, OP13 requires lockstitch equipment:

$$M_{13} = \{\textbf{Lockstitch}\}$$

whereas OP14 requires the hemming/coverstitch machine category:

$$M_{14} = \{\textbf{Hemming}, \textbf{Coverstitch}\}$$

Therefore,

$$M_{13} \cap M_{14} = \emptyset.$$

Although the workstation is numerically balanced, it is technologically infeasible.

LCR contains two machine incompatibilities. The first occurs when OP1 and OP2 are assigned to the same workstation:

$$M_1 \cap M_2 = \emptyset,$$

Because OP1 requires a fusing press and OP2 requires a lockstitch machine. The second incompatibility again concerns OP13 and OP14.

By contrast, RPW-MC and LCR-MC prevent these combinations by evaluating machine compatibility during assignment. For instance, RPW-MC separates OP13 and OP14 into

distinct workstations, eliminating the violation while increasing the workstation count from 14 to 15.

This workstation-level examination demonstrates why the number of workstations and line efficiency alone are insufficient to establish the technological feasibility of a balancing solution.

All evaluated procedures satisfy the precedence requirements:

$$V_P = 0.$$

### 4.3. Comparison with the Exact MILP Reference

The MILP formulation was solved using HiGHS 1.8.0 through the SciPy 1.17.0 milp interface under Python 3.13.5.

For the nominal benchmark, the optimization terminated with the status: Optimal—HiGHS Status 7 and a final relative MIP gap of

$$\textbf{MIP Gap} = 0.000000.$$

The solver returned an optimal objective value of

$$m_{\text{MILP}} = 15$$

workstations after exploring 32 branch-and-bound nodes. The measured wall-clock solution time in the reported execution was approximately 0.94 s.

FOH also requires 15 workstations. Its workstation-count gap from the exact reference is therefore

$$Gap_{\text{FOH}} = \frac{m_{\text{FOH}} - m_{\text{MILP}}}{m_{\text{MILP}}} \times 100 = \frac{15-15}{15} \times 100 = 0\%.$$

Thus, under the nominal operating condition, FOH reaches the minimum feasible workstation count established by the exact model.

Nevertheless, FOH and MILP generate different workstation configurations. The originally reported optimal MILP solution has:

$$SI_{MILP,\,original} = 2.063,$$

Whereas FOH produces

$$SI_{FOH} = 2.763.$$

This difference arises from the allocation of operations among workstations. The original MILP configuration groups OP2 with OP5 and OP3 with OP4, producing two workstation loads of 1.80 min. FOH instead groups OP2 with OP3 and OP5 with OP6, resulting in loads of 1.50 min and 2.00 min, respectively.

The Smoothness Index is calculated relative to the maximum workstation load. Consequently, these differences in workstation composition and load distribution affect the resulting SI values.

Importantly, workload smoothness was not included in the MILP objective function. Therefore, the reported SI of 2.063 characterizes the particular optimal assignment obtained rather than the minimum possible Smoothness Index among all optimal 15-workstation configurations.

An independent computational reconstruction confirmed the minimum feasible requirement of 15 workstations with a zero final MIP gap but returned a different optimal assignment with:

$$SI_{\text{MILP,reconstructed}} = 2.414.$$

This independent execution explored one branch-and-bound node and required approximately 0.247 s. These statistics are distinguished from the original execution, which reported 32 nodes and approximately 0.94 s.

The comparison demonstrates that FOH achieves workstation-count optimality for the nominal benchmark while preserving the modeled precedence, cycle-time, and machine-compatibility requirements through a deterministic constructive procedure. The distinct Smoothness Index values further highlight the importance of evaluating workload distribution separately from workstation-count optimality.

### *4.4. Sensitivity Analysis*

The sensitivity of the comparative results was examined using three cycle-time conditions:

$$\boldsymbol{C} \in \{\mathbf{1.80}, \mathbf{2.00}, \mathbf{2.20}\}\ \mathbf{min},$$

combined with processing-time variations of

$$\boldsymbol{\delta} \in \{-\mathbf{5}\%, \mathbf{0}, +\mathbf{5}\%\}.$$

For each scenario, the adjusted processing time of operation i was calculated as:

$$t_i^{(\delta)} = t_i(\mathbf{1} + \boldsymbol{\delta}).$$

The adjustments were applied uniformly to all operations, whereas precedence relationships and machine requirements remained unchanged. The nine scenarios therefore represent deterministic sensitivity conditions rather than stochastic simulations.

All nine MILP instances were solved to proven optimality with a final MIP gap of zero.

In Table 4, F indicates a feasible configuration and IF denotes a configuration that violates at least one modeled feasibility condition.

FOH remains feasible in all nine tested conditions: **9/9**. SPT produced feasible configurations in seven scenarios 7/9, whereas RPW and LCR were feasible in four scenarios each 4/9. These results confirm the feasibility counts reported in Table 4.

The results also reveal an important effect of cycle-time relaxation. Increasing the available workstation capacity may reduce the number of workstations produced by conventional heuristics, but it also creates additional opportunities to group technologically incompatible operations when machine constraints are not enforced during assignment.

**Table 4.** Sensitivity results for cycle-time and processing-time variations.

| C (min) | Variation | RPW | SPT | LCR | FOH | MILP |
|---|---|---|---|---|---|---|
| 1.80 | −5% | 15/F | 16/F | 15/F | 16/F | 15/F |
| 1.80 | 0% | 15/F | 16/F | 15/F | 16/F | 15/F |
| 1.80 | 5% | 16/F | 16/F | 16/F | 16/F | 16/F |
| 2.00 | −5% | 14/IF | 15/F | 13/IF | 15/F | 14/F |
| 2.00 | 0% | 14/IF | 16/F | 13/IF | 15/F | 15/F |
| 2.00 | 5% | 15/F | 16/F | 15/F | 16/F | 15/F |
| 2.20 | −5% | 12/IF | 12/IF | 11/IF | 14/F | 12/F |
| 2.20 | 0% | 13/IF | 13/IF | 12/IF | 15/F | 13/F |
| 2.20 | 5% | 14/IF | 16/F | 13/IF | 15/F | 15/F |

This effect is particularly visible for:

$$C = 2.20 \text{ min},$$

where RPW and LCR generate numerically compact but machine-infeasible solutions under all three processing-time conditions.

FOH avoids these combinations because the machine-compatibility condition:

$$M_k \cap M_i \neq \emptyset$$

is verified before every task assignment.

However, the sensitivity results also confirm that FOH does not systematically achieve the minimum feasible workstation count. Its relative optimality gap ranges between:

$$0\% \leq \text{Gap}_{\text{FOH}} \leq 16.67\%.$$

FOH matches the exact workstation count in three of the nine tested scenarios.

To establish a comparison under equivalent technological constraints, the compatibility-aware variants RPW-MC, SPT-MC, and LCR-MC were evaluated using the same experimental conditions.

All configurations in Table 5 satisfy the modeled feasibility constraints.

LCR-MC achieves the minimum feasible workstation count in eight scenarios, compared with six for RPW-MC, three for FOH, and one for SPT-MC.

These differences demonstrate that applying identical feasibility constraints does not necessarily produce equivalent workstation configurations. The task-selection strategy remains an important factor affecting workstation utilization.

The largest FOH optimality gap occurs at:

$$C = 2.20 \text{ min}, \quad \delta = -5\%.$$

Under this condition, FOH requires 14 workstations, whereas LCR-MC and MILP require 12.

**Table 5.** Workstation counts obtained by compatibility-aware heuristics under the nine scenarios.

| C (min) | Variation | RPW-MC | SPT-MC | LCR-MC | FOH | MILP |
|---|---|---|---|---|---|---|
| 1.80 | −5% | 15 | 16 | 15 | 16 | 15 |
| 1.80 | 0% | 15 | 16 | 15 | 16 | 15 |
| 1.80 | 5% | 16 | 16 | 16 | 16 | 16 |
| 2.00 | −5% | 15 | 15 | 15 | 15 | 14 |
| 2.00 | 0% | 15 | 16 | 15 | 15 | 15 |
| 2.00 | 5% | 15 | 16 | 15 | 16 | 15 |
| 2.20 | −5% | 14 | 14 | 12 | 14 | 12 |
| 2.20 | 0% | 15 | 15 | 13 | 15 | 13 |
| 2.20 | 5% | 15 | 16 | 15 | 15 | 15 |

The corresponding relative gap is:

$$\text{Gap}_{\text{FOH}} = \frac{14 - 12}{12} \times 100 = 16.67\%.$$

This result illustrates the influence of operation-selection priorities on workstation count. FOH follows a predetermined topological sequence, whereas LCR-MC prioritizes

longer feasible operations, potentially allowing more effective utilization of workstation capacity in certain scenario.

The results therefore support the role of FOH as a deterministic, feasibility-preserving constructive procedure while identifying potential improvements in its task-selection strategy.

The computational behavior of the exact MILP formulation was also examined, particularly when the cycle-time limit increased to 2.20 min. The detailed workstation assignments for each method are reported in Table 3.

The computational performance of the exact MILP formulation across the nine sensitivity scenarios is summarized in Table 6, including solution times, branch-and-bound node counts, and final MIP gaps.

**Table 6.** MILP computational performance under the nine sensitivity scenarios.

| C (min) | Variation | Optimal WS | Time (s) | B&B Nodes | Final MIP Gap |
|---|---|---|---|---|---|
| 1.80 | −5% | 15 | 0.185 | 1 | 0.000000 |
| 1.80 | 0% | 15 | 0.156 | 1 | 0.000000 |
| 1.80 | 5% | 16 | 0.267 | 1 | 0.000000 |
| 2.00 | −5% | 14 | 0.282 | 1 | 0.000000 |
| 2.00 | 0% | 15 | 0.247 | 1 | 0.000000 |
| 2.00 | 5% | 15 | 0.107 | 1 | 0.000000 |
| 2.20 | −5% | 12 | 0.323 | 11 | 0.000000 |
| 2.20 | 0% | 13 | 0.233 | 1 | 0.000000 |
| 2.20 | 5% | 15 | 0.467 | 1 | 0.000000 |

B&B: branch-and-bound. The statistics correspond to the independent MILP reconstruction and must not be attributed to the originally reported solver execution.

At C = 2.20 min, the solver explored 11 branch-and-bound nodes under the −5% processing-time variation, compared with one node for the nominal and +5% conditions.

The corresponding computational times were approximately 0.323, 0.233, and 0.467 s, respectively.

These observations indicate that computational effort does not necessarily vary monotonically with cycle-time relaxation. Increasing workstation capacity changes the combinations of operations that can be assigned together, potentially influencing presolve reductions and branch-and-bound exploration.

However, the reported execution times and node counts are specific to the investigated benchmark and computational environment. They should not be interpreted as evidence of general solver scalability.

Finally, the nine scenarios are deterministic versions of the same 17-operation benchmark rather than nine independent industrial validation cases.

### *4.5. Overall Interpretation*

The results demonstrate that numerical balancing performance and technological feasibility represent distinct but complementary evaluation dimensions.

Under the nominal operating condition, LCR achieves the highest apparent line efficiency of 76.92%, followed by RPW at 71.43%. However, both configurations contain machine-incompatible workstation assignments. Their smaller workstation counts therefore

cannot be interpreted as feasible improvements under the technological assumptions of the model.

FOH generates a technologically feasible configuration with 15 workstations and a line efficiency of 66.67%, matching the minimum workstation count established by the exact MILP formulation. This result demonstrates that the proposed constructive procedure can attain workstation-count optimality for the nominal benchmark while preserving precedence, cycle-time, and machine-compatibility requirements.

The sensitivity analysis reinforces the importance of integrating machine compatibility during workstation construction. Conventional priority-based heuristics may obtain numerically compact configurations when workstation capacity increases, but some of these reductions result from assignments that cannot be implemented using a common admissible machine technology. FOH prevents such combinations through the systematic verification and updating of the admissible machine set after each assignment.

Beyond these numerical findings, the methodological contribution of FOH lies in the explicit organization of task precedence and technological feasibility within a deterministic constructive framework. Its predefined topological sequence establishes a reproducible operation-selection order, and the incremental capacity and machine-compatibility checks maintain the feasibility of each workstation throughout construction. The formal feasibility-preservation property presented in Section 3.3 establishes that, under the stated assumptions, the algorithm terminates with a feasible configuration without requiring subsequent repair of machine-incompatible assignments.

This property should not be interpreted as exclusive to FOH. The compatibility-aware variants RPW-MC, SPT-MC, and LCR-MC also enforce technological feasibility during workstation formation. Their inclusion provides an important methodological distinction between the effect of feasibility enforcement and the influence of the operation-selection strategy.

The comparative results show that the latter remains decisive for workstation-count performance. LCR-MC matches the exact MILP optimum in eight of the nine investigated scenarios, compared with six for RPW-MC, three for FOH, and one for SPT-MC. These differences indicate that alternative selection rules can produce more compact feasible assignments, even when the same technological constraints are applied.

The practical consequences of these differences should also be considered. At a cycle time of 2.20 min under nominal processing times, FOH requires 15 workstations, whereas LCR-MC and MILP require 13. This corresponds to a 15.38% increase relative to the minimum feasible workstation count. Depending on the production organization, the additional workstations may affect staffing, equipment requirements, and operating costs. However, the available benchmark does not provide sufficient industrial cost data to quantify these effects.

The scientific contribution of the present study therefore extends beyond demonstrating the importance of machine-compatibility constraints. It provides a formally specified and reproducible feasibility-oriented construction procedure, establishes its feasibility-preservation property under defined assumptions, and evaluates its behavior against both conventional and compatibility-aware heuristics using an exact optimization reference. This comparative framework makes it possible to distinguish the effect of technological feasibility enforcement from the effect of task-selection priorities on balancing performance.

For the considered garment assembly benchmark, FOH offers a transparent constructive alternative for generating technologically feasible workstation configurations. Its nominal optimality and consistent feasibility across the investigated scenarios support the validity of the proposed mechanism, and the performance of compatibility-aware alternatives identifies opportunities for improving its workstation-count efficiency.

Further validation on independent industrial instances, larger precedence networks, and more heterogeneous machine requirements remains necessary to determine the broader applicability and comparative effectiveness of FOH.

## 5. Study Limitation

The present study has several limitations that should be considered when interpreting the results. First, FOH was evaluated on a single 17-operation garment assembly benchmark. Although nine combinations of cycle-time and processing-time conditions were examined, these scenarios represent deterministic versions of the same precedence network and machine requirements rather than independent industrial validation cases. Additional garment styles, larger instances, and industrial datasets with different technological characteristics are therefore required to assess the generalizability of the findings.

Second, operation processing times were treated as deterministic. The applied five-percent variations provide a controlled sensitivity assessment but do not represent the stochastic fluctuations encountered in garment production. Operator-dependent processing times, machine interruptions, rework, and other disturbances may affect workstation capacity and the feasibility of an initially constructed assignment. Future research should therefore investigate scenario-based and robust extensions of FOH, in which uncertain processing times are explicitly considered during workstation construction and subsequent evaluation.

Worker learning represents another relevant extension. As operators gain experience, task durations may decrease at different rates, potentially changing workstation workloads and the suitability of an initial assignment. Incorporating empirically calibrated learning curves would allow processing-time estimates to evolve with production experience and support periodic rebalancing. Such developments would require longitudinal shop-floor measurements to estimate individual learning parameters and evaluate their effects on workload distribution and production stability.

Third, machine feasibility is represented through admissible machine types, assuming one principal machine technology per workstation. The formulation does not explicitly account for equipment inventory, attachment-change times, or multiple primary machines within the same workstation. Extending the model to these conditions would require additional resource and setup constraints.

Finally, the exact MILP formulation provided proven optimal solutions for all nine scenarios, but this result does not establish its computational performance on larger industrial problems. Additional experiments should investigate larger precedence networks and more extensive technological alternatives. Decomposition-based MILP formulations, matheuristics, and metaheuristic improvement procedures could also be investigated to address increasing computational complexity and improve feasible workstation-count performance.

These limitations define the scope of the present contribution. FOH provides a reproducible framework for enforcing the modeled feasibility conditions, and its broader industrial applicability requires additional validation.

## 6. Conclusions

This study proposed a Feasibility-Oriented Heuristic (FOH) for garment assembly line balancing under precedence, cycle-time, and machine-compatibility constraints. The proposed approach integrates a deterministic topological task-selection sequence with dynamic workstation-capacity verification and the continuous updating of admissible machine sets. This construction mechanism provides a transparent and reproducible proce-

dure for generating technologically feasible workstation configurations under the stated modeling assumptions.

The method was evaluated on a 17-operation garment assembly benchmark and compared with conventional RPW, SPT, and LCR heuristics, their machine-compatibility-aware variants, and an exact Mixed-Integer Linear Programming (MILP) reference. Under the nominal cycle time of 2.00 min, FOH generated a feasible 15-workstation configuration with a line efficiency of 66.67% and no precedence or machine-compatibility violations. This configuration matched the proven MILP optimum, corresponding to a zero workstation-count optimality gap.

The comparative results demonstrated that conventional heuristics may generate numerically attractive but technologically infeasible configurations when machine compatibility is not enforced during assignment. FOH addresses this limitation by incorporating feasibility checks directly into the workstation-construction process. The feasibility-preservation property established in this study further provides a formal justification for the validity of its assignments under the defined assumptions.

Across nine deterministic sensitivity scenarios derived from the same benchmark, FOH maintained feasibility in all cases. Nevertheless, comparison with compatibility-aware classical heuristics revealed that workstation-count performance depends on the task-selection strategy. LCR-MC matched the MILP optimum in eight scenarios, compared with three for FOH, whose maximum relative optimality gap reached 16.67%. These findings highlight the distinction between preserving technological feasibility and minimizing workstation requirements.

The observed differences also have practical implications. At a cycle time of 2.20 min with processing times reduced by 5%, FOH required 14 workstations, compared with 12 for LCR-MC and MILP. Such differences may influence equipment, staffing, and operating requirements, although their financial consequences could not be quantified using the available data.

The principal scientific contribution of this work is the formalization and evaluation of a deterministic feasibility-oriented workstation-construction procedure that explicitly combines topological task ordering, capacity control, and machine-compatibility preservation. By distinguishing the effects of feasibility enforcement from those of task-selection priorities, the study also provides a reproducible comparative basis for assessing constructive garment assembly line balancing methods. FOH is therefore positioned as an interpretable, feasibility-preserving alternative that complements existing priority-based heuristics and exact optimization approaches rather than replacing them.

Future research will focus on improving FOH's task-selection strategy, validating its performance on larger and independent industrial benchmarks, and extending the framework to incorporate limited machine availability, stochastic processing times, worker learning, and operator-specific characteristics.

**Author Contributions:** Conceptualization, Y.E.B. and A.M.; methodology, Y.E.B. and A.M.; software, Y.E.B.; validation, A.M. and A.S.; formal analysis, Y.E.B.; investigation, Y.E.B.; resources, Y.E.B.; data curation, Y.E.B. and A.M.; writing—original draft preparation, Y.E.B.; writing—review and editing, Y.E.B.; visualization, Y.E.B. and A.M.; supervision, A.M. and A.S.; project administration, Y.E.B. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The data supporting the findings of this study are provided within the article, including the benchmark dataset and numerical results presented in Tables 1–6.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References


1. Chen, J.C.; Chen, C.-C.; Su, L.-H.; Wu, H.-B.; Sun, C.-J. Assembly line balancing in garment industry. *Expert Syst. Appl.* **2012**, *39*, 10073–10081. [CrossRef]
2. Bongomin, O.; Mwasiagi, J.I.; Nganyi, E.O.; Nibikora, I. Improvement of garment assembly line efficiency using line balancing technique. *Eng. Rep.* **2020**, *2*, e12157. [CrossRef]
3. Bongomin, O.; Mwasiagi, J.I.; Nganyi, E.O.; Nibikora, I. A complex garment assembly line balancing using simulation-based optimization. *Eng. Rep.* **2020**, *2*, e12258. [CrossRef]
4. Teshome, M.M.; Meles, T.Y.; Yang, C.-L. Productivity improvement through assembly line balancing by using simulation modeling in case of Abay garment industry Gondar. *Heliyon* **2024**, *10*, e23585. [CrossRef] [PubMed]
5. Becker, C.; Scholl, A. A survey on problems and methods in generalized assembly line balancing. *Eur. J. Oper. Res.* **2006**, *168*, 694–715. [CrossRef]
6. Pearce, B.W.; Antani, K.R.; Mears, L.; Funk, K.; Mayorga, M.E.; Kurz, M.E. An effective integer program for a general assembly line balancing problem with parallel workers and additional assignment restrictions. *J. Manuf. Syst.* **2019**, *50*, 180–192. [CrossRef]
7. Pearce, B.W. A Study on General Assembly Line Balancing Modeling Methods and Techniques. Ph.D. Dissertation, Clemson University, Clemson, SC, USA, 2015. Available online: https://open.clemson.edu/all_dissertations/1549/ (accessed on 25 October 2026).
8. Pereira, J. Modelling and solving a cost-oriented resource-constrained multi-model assembly line balancing problem. *Int. J. Prod. Res.* **2018**, *56*, 3994–4016. [CrossRef]
9. Scholl, A.; Becker, C. State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *Eur. J. Oper. Res.* **2006**, *168*, 666–693. [CrossRef]
10. Moon, I.; Logendran, R.; Lee, J. Integrated assembly line balancing with resource restrictions. *Int. J. Prod. Res.* **2009**, *47*, 5525–5541. [CrossRef]
11. Kara, Y.; Atasagun, Y.; Gökçen, H.; Hezer, S.; Demirel, N. An integrated model to incorporate ergonomics and resource restrictions into assembly line balancing. *Int. J. Comput. Integr. Manuf.* **2014**, *27*, 997–1007. [CrossRef]
12. Sime, H.; Jana, P.; Panghal, D. Feasibility of using simulation technique for line balancing in apparel industry. *Procedia Manuf.* **2019**, *30*, 300–307. [CrossRef]
13. Xu, H.; Xu, B.; Yan, J. Balancing apparel assembly lines through adaptive ant colony optimization. *Text. Res. J.* **2019**, *89*, 3677–3691. [CrossRef]
14. Mouloud, A.; El Belghiti, Y.; Tetouani, S.; Soulhi, A. Optimization of order scheduling in the Moroccan garment industry for fast fashion: A clustering-based approach. *Data Metadata* **2024**, *3*, 644. [CrossRef]
15. Virtanen, P.; Gommers, R.; Oliphant, T.E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nat. Methods* **2020**, *17*, 261–272. [CrossRef] [PubMed]
16. Huangfu, Q.; Hall, J.A.J. Parallelizing the dual revised simplex method. *Math. Program. Comput.* **2018**, *10*, 119–142. [CrossRef]


**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.