Energy-aware Configuration of Manufacturing Systems

András Kovács[a,*], Zsombor Szádoczki[a,b], Dávid Karnok[a], Lukas Wittmann[c], Vincent Kalchschmid[c]

[a]*HUN-REN Institute for Computer Science and Control (SZTAKI), Kende u. 13-17, 1111 Budapest, Hungary*
[b]*Department of Operations Research and Actuarial Sciences, Corvinus University of Budapest, 1093 Fővám tér 8., Budapest, Hungary*
[c]*Fraunhofer-Institut für Gießerei-, Composite- und Verarbeitungstechnik (IGCV), Am Technologiezentrum 10, 86159 Augsburg, Germany*

Abstract

This paper addresses the challenge of systematically incorporating detailed energy considerations into early-stage production system design. Unlike existing approaches that either assess energy performance only after a system has been fully specified or rely on coarse consumption estimates during optimization, the proposed methodology integrates detailed energy modeling with mathematical optimization in a novel framework for energy-aware production system configuration. The approach is based on an energy state model that characterizes the consumption of individual system components and production tasks, distinguishing between working and idle states, supporting multiple energy carriers, and capturing energy conversion processes. This model is embedded into a mixed-integer linear programming (MILP) formulation of the configuration problem, which can be solved efficiently using commercial solvers. This enables the simultaneous minimization of investment and energy costs already during early-stage system design. The approach was implemented in a prototype decision-support tool and demonstrated in a case study on a physical production line. Results show that the energy state model provides accurate consumption estimates when extrapo-

*Corresponding author

Email addresses: andras.kovacs@sztaki.hu (András Kovács), zsombor.szadoczki@sztaki.hu (Zsombor Szádoczki), david.karnok@sztaki.hu (Dávid Karnok), lukas.wittmann@igcv.fraunhofer.de (Lukas Wittmann), vincent.kalchschmid@igcv.fraunhofer.de (Vincent Kalchschmid)

lating from measured data of an existing product variant to a future variant with a different process plan. Computational experiments confirm that the approach can solve large-scale configuration problems with up to 12 stations and hundreds of tasks efficiently.

Keywords: Production system configuration, Sustainable manufacturing, Energy consumption, Mathematical programming, Optimization

---

1. Introduction

Escalating and increasingly volatile energy prices [1], together with carbon footprint reporting obligations and performance improvement requirements driven by regulatory and market forces, are turning energy use into a critical yet high-risk area for manufacturing companies [2, 3]. Since the energy consumption of a production system is largely determined during the early stages of the system design process, energy-aware production system configuration is attracting growing attention from both academia and industry [4, 5]. Although methodologies exist for modeling the energy consumption of fully specified systems [6], as well as for configuring systems to minimize energy costs based on coarse consumption estimates [7], defining formal optimization approaches for energy-aware system configuration that capture the full complexity of energy use in manufacturing systems remains an open challenge.

In order to address this gap, this paper proposes a systematic approach to the energy-aware configuration of assembly lines. Given the description of the products and their corresponding process plans, the system configuration problem involves selecting an appropriate combination of manufacturing system components and assigning production tasks to them while satisfying all technological and productivity requirements. In contrast to conventional methods that focus solely on minimizing investment costs, the proposed approach enables the accurate prediction of energy consumption already at the system configuration stage, and minimizes the combination of total investment and energy costs over the planning horizon. To this end, a sophisticated energy state model, based on the Energy Block concept of [8], is introduced to characterize energy consumption at the level of individual components and tasks. The model captures complex interactions among system components, accounts for multiple energy carriers, and represents conversions between different carriers. This energy state model is integrated into a mathematical

programming formulation of the production system configuration problem, enabling the computation of exact optimal solutions for the energy-aware configuration problem. The proposed techniques have been implemented in a prototype decision-support software tool that facilitates the configuration of production systems through an interactive workflow combining automated optimization with manual refinement steps. The presented techniques have been validated from multiple perspectives, including modeling capabilities, the accuracy of predicted energy consumption, and computational efficiency.

The remainder of the paper is structured as follows. First, Section 2 provides a review of the relevant literature. Section 3 formally defines the production system configuration problem. Next, Section 4 introduces the energy state model used to characterize consumption, followed by a mathematical programming formulation of the configuration problem in Section 5. Section 6 presents the implementation of the proposed techniques within a decision-support software prototype. The proposed approach is validated in a series of experiments in Section 7. Finally, Section 8 concludes the paper and outlines directions for future research.

2. Literature Review

2.1. Modeling the Energy Consumption of Manufacturing Systems

Energy-aware design and operation of manufacturing systems rely on the availability of suitable energy models that describe the relationship between production activities and energy consumption. Manufacturing systems typically comprise multiple stations, machines, and subsystems, as well as multiple energy carriers whose consumption patterns vary over time according to production requirements. Energy consumption is affected by numerous factors, including machine states, process execution, production volume, and system utilization [9]. Capturing these interactions in a model is challenging because a trade-off exists between representational accuracy and model complexity. Consequently, researchers have proposed a variety of energy modeling approaches, ranging from high-level estimation methods to detailed physics-based and data-driven models, each supporting different planning and optimization approaches.

Advanced energy modeling methodologies have been developed to capture both multi-domain physical interactions and cross-level dependencies in manufacturing systems. At the equipment level, bond graph theory has been employed to integrate mechanical transmission dynamics with energy flows,

enabling detailed analysis of energy conversion and dissipation within complex subsystems such as machine tool spindles [10]. To address interactions across different organizational scales, cross-level modeling frameworks have been proposed to link local process behavior with production-line dynamics [11]. These approaches embed machine-level optimization models, such as Pareto-optimal energy–time trade-off functions, into higher-level stochastic representations, thereby capturing the effects of buffering, resource interactions, and production disruptions on overall energy performance. Detailed, process-level models for evaluating performance and sustainability, capturing stochastic system dynamics, often rely on Markov chain models [12]. The extension of such models to multi-process systems, specifically, hybrid additive-subtractive manufacturing systems, is investigated in [13].

A widely adopted approach to energy modeling is based on the work of [8]. In this framework, energy consumption data are first obtained empirically, after which the relationship between a system's operating status (OS) and its energy consumption is established. For instance, the power consumption of a production resource is typically higher during value-adding operations than during idle periods. Depending on the complexity of the system and the level of process aggregation, multiple operating statuses and transitions between them may be distinguished. By associating each OS with a representative energy consumption, an energy state (ES) model can be constructed that captures the dynamic energy behavior of the system.

Research on ES models has been dominated by applications to machine tools (e.g., [14, 15, 16]). Although the resulting models differ in their level of detail and application domain, they follow a common principle: representing system behavior as a sequence of discrete operating states with characteristic energy consumption. The surveyed models typically distinguish between three and eight energy states, including, in various combinations, key operating states such as *working*, *idle*, *setup*, *error*, and *off*. For the early-stage design of high-volume manufacturing systems, however, setup and error states are often neglected, making the *working* and *idle* states the most relevant.

2.2. Configuration of Manufacturing Systems

Production system configuration concerns the translation of product specifications and process plans into production system design: it involves selecting system components and assigning production tasks to them. These decisions fundamentally determine the productivity, cost, and energy consumption of the resulting system. A plethora of different models and solution

approaches has been investigated in the scientific literature.

In case of serial production systems, the archetypal mathematical formulation is the assembly line balancing problem (ALBP), in which tasks are assigned to stations either to minimize the number of stations subject to a given cycle time (SALBP-1) or to minimize the cycle time for a fixed number of stations (SALBP-2). A taxonomy of ALBP variants is provided in [17], while solution approaches are reviewed in [18]; recent advances are summarized in the comprehensive survey [19]. Several extensions, such as *Robotic* ALBP [20] and *Resource-dependent* ALBP [21] combine line balancing with resource selection, which enables the consideration of alternative processing modes and their implications for task durations, energy consumption, and system costs.

Various attempts have been made to capture the minimization of energy consumption within ALBP. In [22], it is argued that energy use can be reduced indirectly by minimizing system size and cycle time, i.e., via classical ALBP objectives, without explicitly modeling energy consumption. Among approaches that model energy consumption explicitly, many assign fixed consumption values to each task–robot assignment. These include [23], which minimizes cycle time and total energy consumption using particle swarm optimization; [24], which proposes a constraint programming model for balancing human–robot collaborative assembly lines while incorporating a wide range of practical constraints and cost factors, including working-state robot energy consumption; [7], which minimizes cycle time and energy consumption using a multi-criteria non-dominated sorting genetic algorithm (NSGA); and [25], which addresses a mixed-model sequencing and assembly line balancing problem by minimizing a linear combination of cycle time and energy consumption via a hybrid of ant colony optimization and simulated annealing.

More sophisticated models of energy consumption distinguish between robots' working and idle states. These include [26], who compute Pareto-optimal solutions minimizing cycle time and energy consumption using simulated annealing; [27], who propose a multi-objective cooperative coevolutionary algorithm to minimize carbon footprint, directly proportional to robot energy consumption, and maximize line efficiency; [28], who developed a Pareto artificial bee colony algorithm for minimizing cycle time and energy consumption; [29], who minimize energy consumption in mixed-model assembly lines using cuckoo search; and [30], who combine multi-objective harmony search with reinforcement learning to solve a line balancing problem using a

rich process model capturing real-world requirements of automated electronic assembly lines. A similar problem in mixed-model, multi-line human–robot collaborative systems is addressed in [31], where total production costs and emissions from energy consumption are minimized using NSGA. A problem called *sustainable line balancing*, jointly minimizing cycle time, energy consumption, and ergonomic risk, is introduced in [32], with Pareto-optimal solutions obtained via a hybrid of Q-learning and a memetic algorithm. A more detailed energy consumption model is considered in [33], which distinguishes between electricity and gas usage of workstations and robots; the authors employ a hybrid large neighborhood search and genetic algorithm to derive the Pareto set for multiple cost- and sustainability-related criteria in a stochastic automotive disassembly line balancing problem.

Remotely related to energy consumption in automated systems, integrating human energy expenditure in manned assembly systems into ALBP models received considerable attention, too, as a means for ensuring ergonomics, see, e.g., [34].

Energy consumption can also be addressed at other levels of the enterprise resource hierarchy. At the equipment level, [35] propose a data-driven method to predict and optimize the energy use of industrial robots along specified trajectories, while [36] present an approach to reduce machine tool energy consumption through appropriate selection of machining parameters. Reviews of energy-aware scheduling in intelligent production systems are provided in [37] and [38], and a recent survey on energy-flexible manufacturing systems is given in [39].

2.3. Contribution

This section positions the contribution of this paper with respect to two distinct lines of research: (1) energy and sustainability evaluation techniques, and (2) previous approaches to energy-aware production system configuration.

Evaluation techniques enable investigating the impact of system configuration and process parameters on productivity and sustainability using sophisticated models, based on candidate system configurations provided as input, thereby supporting the improvement of manufacturing system performance. However, they do not directly compute a system configuration that optimizes a selected performance measure. Integrating an energy consumption model into an optimization approach inevitably involves a trade-off between prediction accuracy and computational tractability. Energy state

models, which simplify temporal dynamics by representing energy consumption as stationary energy blocks, offer a promising means of maintaining computational tractability even when coupled with the detailed model of technological dependencies required for system configuration.

Earlier energy-aware production system configuration approaches, including the above line balancing methods, work on strongly simplified models of the production system, while they neglect their complex and interconnected technological and energy requirements. This paper substantially extends those approaches by incorporating the following important features:

- Characterization of the technological requirements of tasks and components using a rich model of function requirements, including symbolic as well as additive and maxitive numerical functions.
- Automatic inclusion of auxiliary components in the configuration, as implied by the function requirements. This is particularly important because auxiliary components often account for a substantial share of the total energy consumption of production systems.
- A detailed energy state model that differentiates between the working-state and idle energy consumption of components.
- Modeling multiple energy carriers in the production system.
- Modeling energy conversion within the production system through the automatic selection and inclusion of suitable energy converter components.
- Automatically computing a system configuration that minimizes a combination of investment and energy costs using exact mathematical programming methods.

A precursor of the model presented here was introduced in [40], focusing on an explainable optimization approach to system configuration, but without considering energy consumption. Earlier works [41, 42] applied similar models to manage fluctuations arising from machine malfunctions and the variability of human operation times. A comparable production system configuration model for flexible manufacturing cells was proposed in [43], with particular emphasis on the integration of system configuration and geometric layout planning.

3. Problem Definition

In this section, the energy-aware production system configuration problem is first motivated by a practical example. A formal problem definition is presented afterwards.

3.1. A Motivating Sample Problem

Throughout the paper, the different aspects of production system configuration are illustrated using a sample use case based on the Bosch Rexroth modular Mechatronic System (mMS) 4.0 (see Figure 1) [44]. This system was selected due to its accessibility for experiments involving different operating modes and speeds, as well as for energy consumption measurements. Furthermore, the complexity of the demonstration system makes it suitable for presenting the core features of the proposed approach without introducing unnecessary complications.

The mMS line consists of three stations and performs the assembly of a cubic product from two parts. In the first station (feeding station), the two parts are loaded sequentially onto a conveyor, where inspection tasks are also carried out. The parts are then transferred to the second station (assembly station), where joining pins are inserted into the lower part, followed by the pressing task that assembles the two parts. Finally, the assembled product is stored in the high-bay warehouse of the third station. The complete assembly process consists of 17 tasks, as illustrated in the Gantt chart shown in Figure 2.

Given a formal specification of the process plan, a library of candidate system components, and a cycle time limit (45 s in the example, corresponding to approximately 100 000 pieces per year under single-shift production with typical OEE), the production system configuration problem involves assigning components to tasks and installing all required auxiliary components in the line in the most efficient manner. Candidate components differ in capabilities, processing speed, cost, and required energy carriers (e.g., electricity, compressed air, or both). Selecting the most appropriate component for each position requires considering several factors, including:

- Each task must be assigned to a component with appropriate capabilities.
- Installing a component may generate additional function requirements, which in turn necessitate auxiliary components (e.g., every pneumatic

Figure 1: The Bosch Rexroth modular Mechatronic System (mMS) 4.0 [44].

component requires compressed air, which must be provided by a dedicated air compressor if compressed air is not supplied externally).

- The processing speed of the selected component must support the desired cycle time. However, whether the cycle time limit is strict in a given station depends on the components selected for other tasks and the synchronization constraints among them.

- The required energy carriers significantly influence component choice. Pneumatic components (e.g., presses or screwdrivers) are often cheaper than their electric alternatives, but while electricity is ubiquitous, compressed air may not be. Installing pneumatic components may therefore necessitate a dedicated air compressor, which is justified only if multiple pneumatic components are present.

- Although pneumatic components generally have lower purchase costs, their compressed air consumption incurs operational costs. Considering both investment and energy costs may make higher-cost electric components more advantageous.

- The decision maker has flexibility in determining the line architecture. For instance, the entire line could be controlled by a single PLC, requiring a more powerful unit with many I/O channels, or individual

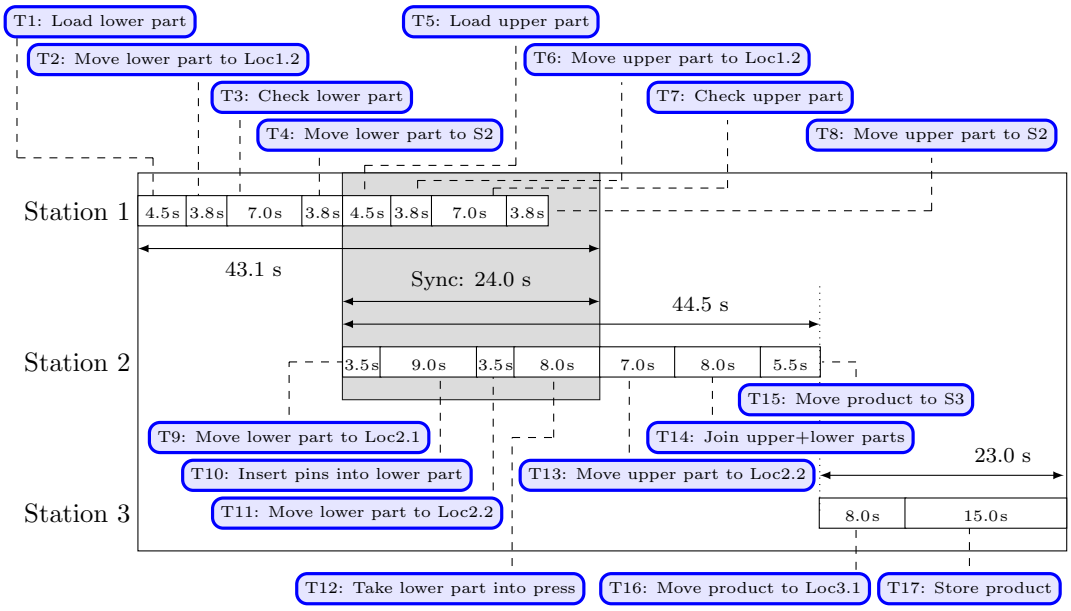

Figure 2: Gantt chart of the tasks executed in the mMS system. Tasks 5-8 in Station 1 must be synchronized with Tasks 9-12 in Station 2. The cycle time of 44.5 s of the overall line is determined by the slowest station, Station 2.

stations could be controlled by dedicated PLCs, allowing the use of smaller, less expensive units.

These considerations are mutually interdependent, giving rise to a complex optimization problem, which is formalized in the next section.

3.2. Formal Problem Definition

The studied production system configuration problem concerns the design of a single serial *production line* for manufacturing multiple products in accordance with their prescribed process plans. For each product, the process plan specifies a fully ordered sequence of *tasks*. The overall structure of the line is predefined by the customer and provided as input: it consists of a series of *stations* arranged along a conveyor, where tasks must be performed at designated *locations* within each station. This structure induces a natural three-level hierarchy: locations within stations, stations within the line, and the line as a whole. This hierarchy is depicted in Figure 3.

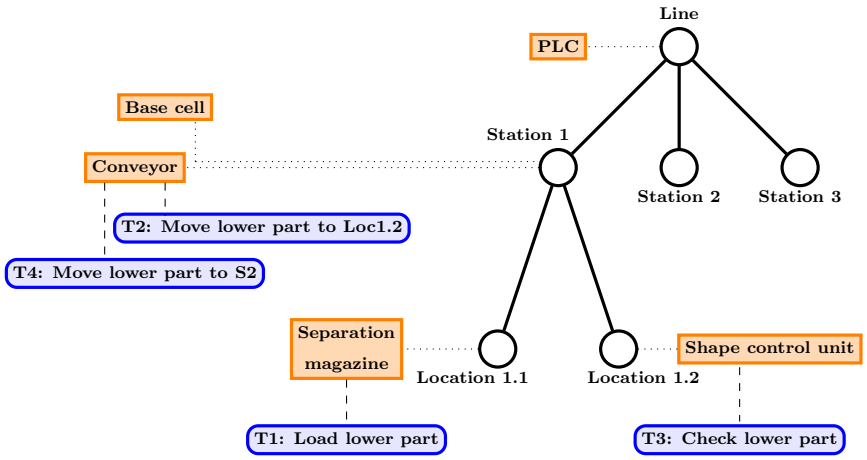

Figure 3: Three-level resource hierarchy, with line, stations, and locations (black nodes). Components (orange labels) are installed at nodes, and tasks (blue, rounded labels) are assigned to components. Only a part of the nodes, components, and tasks are shown.

Tasks, components, and function requirements. The core challenge lies in selecting and deploying, across different nodes of the resource hierarchy, the most suitable combination of components from a predefined component library to manufacture the target products as efficiently as possible. To this end, the technological requirements of tasks are expressed via *function requirements*: for example, a screw-driving task demands the screw-driving function. This function requirement can be satisfied by choosing an appropriate component from the library that provides the needed function, assigning it to the task, and installing it at an appropriate node in the system. For example, the screw-driving function requirement may be satisfied by either an electric or a pneumatic screwdriver.

Similarly, components themselves may have their function requirements. For instance, the pneumatic screwdriver requires the compressed air function, which must be provided by an air compressor. Every function requirement, whether arising from a task or from a component, must be fulfilled by another component installed either at the same node as the origin of the requirement, or at another node above the origin of the requirement in the system hierarchy. For example, if a pneumatic screwdriver is assigned to a screw-driving task, its need for compressed air can be met by an air compressor located at the task location, the corresponding station, or the overall production

line. Installing the compressor at the line level is often advantageous, as it can then serve all components across the entire line. For each component in the library, the allowed levels of installation—location, station, or line—are predefined.

Types of function requirements. Technological requirements of different natures can be encoded into *symbolic* and *numeric* functions. A symbolic function is either required (provided) by a task (component) or not. In the case of numeric functions, tasks and components may require and provide different amounts of the function. Numeric functions are further subdivided into additive functions, where, to satisfy the requirements of multiple tasks and components, the *sum* of the required amounts must be provided (see Figure 4); and *maxitive* functions, where the maximum of the individual requirements must be supplied (see Figure 5). For example, the screw-driving function above is a symbolic function, since it is either required (provided) or not. PLC channels for component control are additive, since the number of channels required from the PLC is the sum of the requirements of the individual components being controlled. Compressed air pressure can be regarded as a maxitive function, since if a compressor provides the maximum of the required pressure values, then it can serve all individual components, potentially through a pressure-limiting valve. Since symbolic functions can be regarded as a special case of maxitive functions, where all required and provided amounts equal one, in the sequel we resort to modeling additive and maxitive functions.

Energy consumption, supply and conversion. A special goal of the system configuration problem is capturing the energy consumption of production already during the time of line design. For this purpose, we adopt a simplified variant of the energy state model defined in [8]. The model captures multiple energy carriers, e.g., electricity and compressed air, and assumes that consumption is composed of (1) the working-state consumption of components working on tasks, which depends both on the component and the task; and (2) the idle consumption of components during time periods where they do not work on tasks.

While some energy carriers are supplied externally at given costs, others may need to be generated internally within the system by installing appropriate energy converter components. For example, if electricity is supplied externally but compressed air is not, and pneumatic components are installed in the system, then an air compressor must also be acquired.

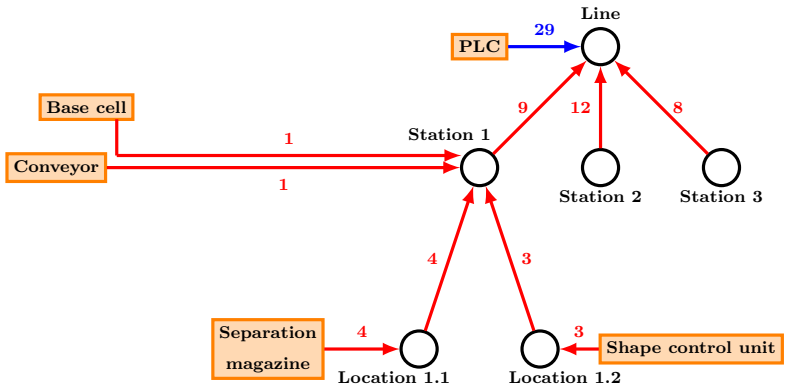

Figure 4: Example of an additive numeric function: PLC channels. The amount required by the components is summarized in the individual nodes and propagated upwards to the root (line-level) node, where the requirement is satisfied by a PLC component.

Function groups. In order to capture rich technological requirements that must be characterized using several numeric parameters, the notion of *function groups* is introduced to express the constraint that certain subsets of functions and energy carriers must be provided together by the same component. A typical example is the combination of the compressed air pressure function and the compressed air energy carrier, which must be supplied by the same air compressor. Hence, a single compressor that provides both high pressure and efficient generation of air cannot be replaced by two separate compressors, one producing high pressure and the other producing low-cost air.

Cycle times and synchronization. The input also specifies a target cycle time for each product. The overall line cycle time is determined by the slowest station, while the cycle time of an individual station is calculated as the sum of the durations of the tasks performed sequentially on the product at that station. Task durations vary depending on the component assigned. Additionally, synchronization constraints are imposed between certain pairs of stations to account for the transfer of multiple parts at specific moments within a cycle. These moments define synchronization points, which require that the total duration of two corresponding subsets of tasks at the two neighboring stations be equal. An example of synchronization is presented in Figure 2. In Station 1, Tasks 1-4 are performed on the lower part, after which

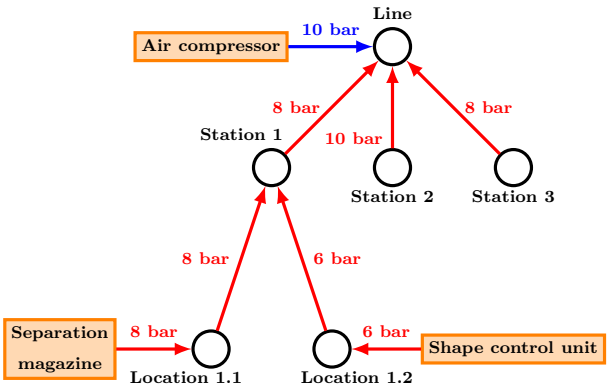

Figure 5: Example of a maxitive numeric function: compressed air pressure. In each node, the maximum of the amounts required by components is taken, and then propagated upwards to the root (line-level) node, where the requirement is satisfied by an air compressor.

this part is transferred to Station 2. Then, Station 1 executes inspection on the upper part as well (Tasks 5-8), and in parallel with that, Station 2 prepares the lower part for assembly (Tasks 9-12). The upper part is transferred to Station 2 after the completion of both Task 8 in Station 1 and Task 12 in Station 2. This implies a synchronization constraint, stating that the total durations of Tasks 5-8 and Tasks 9-12 must be equal.

Objectives. The production system configuration problem involves determining the components installed at the different nodes of the resource hierarchy, as well as the component assigned to each task, in such a way that all the above constraints are satisfied and the cost of the system is minimized. The total cost is composed of the investment cost of the components installed at the nodes and the estimated energy consumption of the system over the planning horizon, during which given amounts of different products are expected to be manufactured.

Formal Domain Metamodel. Synthesizing the technological, topological, and energetic concepts detailed above, the configuration domain is formally defined as an algebraic tuple $\mathcal{M} = \langle \mathcal{R}, \mathcal{T}, \mathcal{C}, \mathcal{F}, \mathcal{E} \rangle$:

- **Resource Topology ($\mathcal{R}$):** A three-level rooted tree $\mathcal{R} = (\mathcal{V}, \mathcal{E}_{\mathcal{V}})$ representing the resource hierarchy, where the root node $v_0 \in \mathcal{V}$ denotes

the entire production line, intermediate nodes stand for stations, and leaf nodes denote individual locations, see Figure 3.

- **Tasks ($\mathcal{T}$):** Each task $t \in \mathcal{T}$ is defined by its target execution node $v_t \in \mathcal{V}$, its function requirement vector $\mathbf{r}_t^T = [r_{f,t}^T]_{f \in \mathcal{F}}$, and its baseline duration vector $\mathbf{d}_t = [d_{t,c}]_{c \in \mathcal{C}}$ across compatible components.

- **Component Library ($\mathcal{C}$):** Each candidate component $c \in \mathcal{C}$ is characterized by its investment cost k_c^C, allowable deployment nodes $\mathcal{V}_c^{\text{allowed}} \subseteq \mathcal{V}$, provided functions $\mathbf{q}_c^C = [q_{f,c}^C]_{f \in \mathcal{F}}$, induced auxiliary function requirements $\mathbf{r}_c^C = [r_{f,c}^C]_{f \in \mathcal{F}}$, and power consumption $\mathbf{p}_c^I = [p_{e,c}^I]_{e \in \mathcal{E}}$ for the idle state and $\mathbf{p}_c^W = [p_{e,c,t}^W]_{e \in \mathcal{E},\ t \in \mathcal{T}}$ for working state across energy carriers and tasks.

- **Function Space ($\mathcal{F}$):** Partitioned into additive functions $\mathcal{F}_A$ (accumulated via summation) and maxitive functions $\mathcal{F}_M$ (evaluated via the maximum operator).

- **Energy Carriers ($\mathcal{E}$):** Partitioned into externally supplied and internally converted carriers. An auxiliary converter component $c \in \mathcal{C}$ converts carrier $E_{\text{from}}(c)$ to $E_{\text{to}}(c)$ via an affine transfer function.

Assumptions. The following list summarizes the assumptions made in the model:

- The model addresses the configuration of a single serial production line.

- Large-series production is assumed; hence, changeovers and ramp-up/ramp-down effects are neglected.

- No in-process buffers are considered.

- The possibility of resource failures is omitted.

- The logical architecture of the line is represented by a three-level hierarchy comprising locations, stations, and the line itself.

- For each product, the predefined process plan specifies a fully ordered sequence of tasks.

- Each task is assigned to a predefined node of the resource hierarchy.

- Components must be selected from a given library, and installed at the nodes of the resource hierarchy.
- Technological correctness is captured through function requirements. Functions are required by tasks and components, and they are either provided by components or supplied externally.
- A function requirement can only be fulfilled by a component installed at the same node or at an ancestor node in the resource hierarchy.
- Functions are additive (the total required amount is the sum of the individual requirements) or maxitive (the total required amount is the maximum of the individual requirements). Symbolic functions are treated as a special case of maxitive functions.
- The model captures multiple energy carriers, which are either supplied externally at given costs or generated within the system by appropriate energy converters.
- Energy consumption comprises the working-state consumption of task–component pairs and the idle-state consumption of components.
- The expected production volumes of the different products over the planning horizon are given, and energy-related operational costs are evaluated over this horizon.
- The model focuses exclusively on production system configuration; downstream planning problems, such as detailed scheduling, are out of scope. Consequently, performance measures that depend on the detailed task schedule, such as peak loads, cannot be captured.
- All model parameters are deterministic.
- The objective is to minimize a linear combination of investment and energy costs over the given planning horizon, defined by the foreseen production volumes of the individual products. With unit weights, this equals the total investment and energy costs.

4. Modeling the Energy Consumption

As indicated previously, the characterization of consumption is based on the Energy Block approach [8], in which consumption is assigned to individual components of the production system (see Figure 6). Components may take different *energy states* over time depending on their activity. Focusing on mass production with large batch sizes, this paper distinguishes two energy states: the *working state*, where the component executes a specific task and consumption depends both on the component and the task; and the *idle* state, where no task is actively executed by the specific component, and hence, the consumption depends solely on the component.

It is emphasized that for each task, only the component explicitly assigned to the task in the configuration model has working state consumption related to the specific task. The duration of the working state equals the duration of the task. Within an energy state, the consumption intensity of the related component is assumed to be constant and given in the input, see Figure 6.

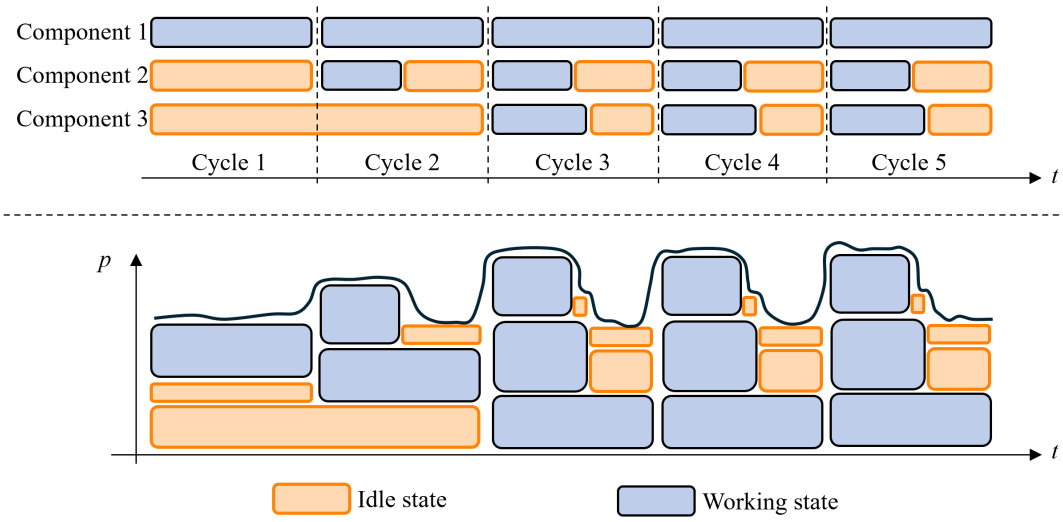

Figure 6: Illustration of the Energy Blocks method for modeling the working state and idle state of production system components. The top section shows the states of the individual components during different cycles, while the bottom section shows the resulting total power requirement of the system. Cycles 1 and 2 represent the system's ramp-up phase, while cycles 3 through 5 represent series production itself.

The propagation of energy consumption upwards in the resource hierarchy is based on an idea similar to additive functions: the total energy demand in a node equals the consumption of the components installed at the specific node in addition to the demand propagated upwards from child nodes. In the applied model, all consumption values represent the total consumption over the planning horizon, taking into account the planned amounts of goods produced.

The model captures an arbitrary number of energy carriers. For instance, in the running example, electricity and compressed air are considered. The demands and supplies are computed separately for each carrier.

The model differentiates *externally supplied* and *internally generated* energy carriers. For example, in the mMS example, external supply is required for electricity, whereas compressed air is generated within the system by an air compressor. For externally supplied carriers, the amount propagated upwards by the line-level node equals the amount consumed by the overall line, and this demand is satisfied by purchasing the energy carrier at the specified unit cost. Otherwise, i.e., if the carrier is not supplied externally or it is less expensive to generate it within the system than to purchase it directly, the carrier can be generated by installing an appropriate energy converter component in a selected node, from where it can satisfy the demand of downstream nodes.

Energy conversion is modeled using a linear transfer function. Consider a converter component c (e.g., an air compressor) that generates energy carrier $E_{\text{to}}(c)$ (compressed air) from energy carrier $E_{\text{from}}(c)$ (electricity). To generate σ units of $E_{\text{to}}(c)$ over a time interval τ, the converter consumes $a_c\sigma + b_c\tau$ units of $E_{\text{from}}(c)$, where the first term, proportional to the generated amount, represents the conversion efficiency, while the second term captures the converter's idle energy consumption.

Figure 7 presents the energy consumption of the mMS line belonging to the production of a single workpiece. Compressed air is generated from electricity within the system by an air compressor installed at the line level, while electricity is supplied externally.

5. Solution Approach

The energy-aware system configuration problem is modeled as a mixed-integer linear program (MILP). Due to the complexity of the model, the presentation is broken into three parts: the core of the system configuration

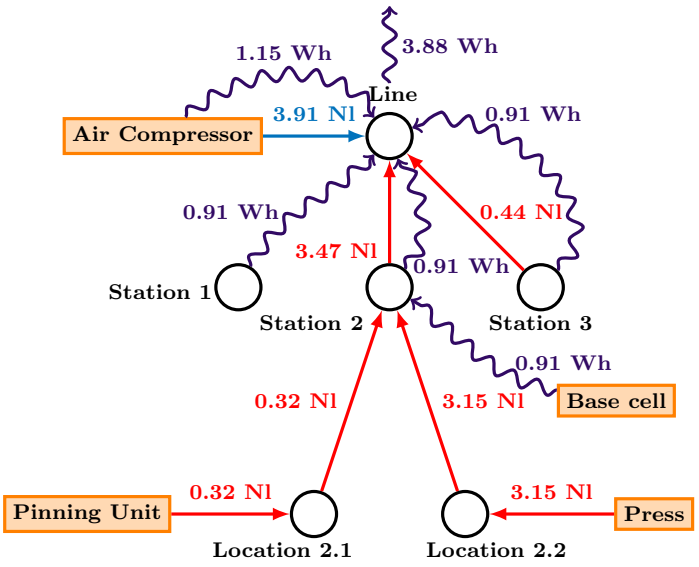

Figure 7: Illustration of the energy consumption model. The amount of compressed air required by the components (red arrows) is summarized in the individual nodes and propagated upwards until the root (line-level) node, where the requirement is satisfied by an air compressor via conversion from electricity. The amount of electricity consumption (dark purple wavy arrows) is handled in a similar way; however, this energy carrier is supplied externally, thus, the line-level node propagates the demand upwards. Only a part of the nodes and components are shown. The values align with the production of a single workpiece of the mMS line detailed in Sections 3.1 and 7.1.

model (Section 5.1), the modeling of function requirements (Section 5.2), and the modeling of energy consumption (Section 5.3). The overall model is recapitulated in Section 5.4. The formal notation is introduced in Tables 1 and 2.

It is noted that the actual implementation contains additional constraints to capture some practical requirements, e.g., fixing components at given locations or fixing the assignment of a component to a task based on the user's request specified in the graphical user interface. Such auxiliary constraints are mathematically straightforward, and they are omitted here for the sake of relative brevity.

5.1. Core of the System Configuration Model

The core of the system configuration model is presented below. This core defines the resource hierarchy, the assignment of tasks to components, and

Table 1: Notation: indices and parameters.

Indices	
p	Product index
t	Task index
c	Component type index
s	Station index
l	Location index
f	Function index
e	Energy carrier index
$p(t)$	Product belonging to task t
$s(t)$	Station assigned to task t
$l(t)$	Location assigned to task t
Input parameters	
$\mathcal{C}(s,l)$	Set of child nodes of station s and location l in the resource hierarchy
$d_{t,c}$	Baseline duration of task t when executed by component c
T_p	Cycle time limit of product p
$\mathcal{S} = \{(S_1, S_2)\}$	Set of synchronization constraints between two subsets of tasks
Λ	Set of components applicable on line level
Σ	Set of components applicable on station level
Π	Set of components applicable on location level
k_c^C	Investment cost of component c
$\mathcal{FA}$	Set of additive functions
$\mathcal{FM}$	Set of maxitive functions
$r_{f,t}^T$	Amount of function f required by task t
$r_{f,c}^C$	Amount of function f required by component c
$q_{f,c}^C$	Amount of function f provided by component c
q_f^X	Maximum amount of function f supplied externally
$\mathcal{EX}$	Set of energy carriers supplied externally
$p_{e,c}^I$	Power consumption of carrier e at idle of component c
$p_{e,c,t}^W$	Power consumption of carrier e of component c when executing task t
$E_{\text{to}}(c)$	Energy carrier generated by energy converter component c
$E_{\text{from}}(c)$	Energy carrier consumed by energy converter component c
a_c, b_c	Parameters of the linear transfer function of energy converter component c
k_e^E	Unit cost of energy carrier e
$\mathcal{G}^{FF}, \mathcal{G}^{EE}, \mathcal{G}^{FE}$	Sets of function-function, energy carrier-energy carrier, and function-energy carrier pairs combined into a function group, respectively
V_p	Units of product p produced over the planning horizon
w_I	Weight of investment cost in the objective function
w_E	Weight of energy cost in the objective function

the cycle times of the different products.

In the representation of the three-level resource hierarchy, nodes are identified by a pair (s, l), where s is the station index and l is the location index.

Table 2: Notation: decision variables.

Core decision variables	
$x_{c,s,l}$	Component c is installed at location l of station s (boolean)
$y_{t,c,s,l}$	Task t is assigned to component c at location l of station s (boolean)
δ_t	Actual duration of task t (real)
τ_p	Actual cycle time of product p (real)
Decision variables related to function requirements	
$\varphi^R_{f,s,l}$	Amount of function f required at location l of station s (real)
$\varphi^U_{f,s,l}$	Amount of function requirement f propagated upwards from location l of station s (real)
$\varphi^P_{f,s,l}$	Function f is provided at location l of station s (boolean)
$\varphi^C_{f,s,l,c}$	Function f is provided by component c at location l of station s (boolean)
Decision variables related to energy consumption	
$\sigma^R_{e,s,l}$	Amount of energy carrier e required at location l of station s (real)
$\sigma^U_{e,s,l}$	Amount of energy carrier requirement e propagated upwards from location l of station s (real)
$\sigma^G_{e,s,l,c}$	Amount of energy carrier e converted into another carrier by component c at location l of station s (real)
$\sigma^P_{e,s,l}$	Energy carrier e is provided at location l of station s (boolean)
$\sigma^C_{e,s,l,c}$	Energy carrier e is provided by component c at location l of station s (boolean)
$\sigma^I_{e,s,l,c}$	Idle consumption of energy carrier e by component c at location l of station s (real)

In case of the root node corresponding to the line itself, $s = 0$ and $l = 0$. For station nodes, $s \geq 1$ and $l = 0$, whereas for location nodes, both $s \geq 1$ and $l \geq 1$. Accordingly, $(0, l)$ with $l \geq 1$ is an invalid node identifier, since locations do not exist outside stations. The set of children $\mathcal{C}(s, l)$ of node (s, l) is composed of the following nodes:

- for the line node $(0, 0)$, the set contains all station nodes $(s, 0)$ with $s \geq 1$;
- for station nodes $(s, 0),\ s \geq 1$, it comprises all location nodes (s, l) with $l \geq 1$.
- whereas for location nodes $(s, l),\ l \geq 1$, the set is empty.

In the MILP model of the configuration problem, the objective is to minimize a linear combination of the investment cost and the energy cost incurred over the planning horizon (1). Here, $\sigma^U_{e,0,0}$ corresponds to the energy demand of carrier e propagated upwards from the line node of the resource hierarchy for the ensemble of all products manufactured during the planning horizon,

i.e., the demand that is not satisfied by generation within the line, but must be purchased from external sources. Constraint (2) ensures that every task is assigned to exactly one component that is located appropriately and provides all the functions required by the task. Inequality (3) encodes that a component can be assigned to a task only if it is installed. Constraints (4), (5) and (6) state that components can be installed only on the appropriate levels of the resource hierarchy, i.e., on the line, on the station, and on the location levels, respectively. Line (7) excludes installing components at invalid nodes, i.e., locations outside stations. Inequality (8) defines the actual durations of tasks depending on the component selected for their execution. It is noted that the actual duration can be higher than the baseline duration due to synchronization constraints between two subsets of tasks (9). Inequalities (10) define product cycle times and state that they must respect the given upper bound at each station. Finally, line (11) defines the binary variables in the model.

minimize

$$w_I \sum_{c,s,l} k_c^C x_{c,s,l} \;+\; w_E \sum_{e} k_e^E \sigma_{e,0,0}^U \tag{1}$$

subject to

$$\sum_{\substack{s \in \{0,\ s(t)\} \\ l \in \{0,\ l(t)\} \\ c:\ r_{f,t}^T \le q_{f,c}^C\ \forall f}} y_{t,c,s,l} = 1 \qquad \forall t \tag{2}$$

$$y_{t,c,s,l} \le x_{c,s,l} \qquad \forall t, c, s, l \tag{3}$$

$$x_{c,0,0} = 0 \qquad \forall i \notin \Lambda \tag{4}$$

$$x_{c,s,0} = 0 \qquad \forall i \notin \Sigma \tag{5}$$

$$x_{c,s,l} = 0 \qquad \forall i \notin \Pi \tag{6}$$

$$x_{c,0,l} = 0 \qquad \forall c, l \ge 1 \tag{7}$$

$$\delta_t \ge \sum_{c,s,l} d_{t,c} y_{t,c,s,l} \qquad \forall t \tag{8}$$

$$\sum_{t \in S_1} \delta_t = \sum_{t \in S_2} \delta_t \qquad \forall (S_1, S_2) \in \mathcal{S} \tag{9}$$

$$T_p \;\ge\; \tau_p \;\ge \sum_{t:\ s(t)=s\ \wedge\ p(t)=p} \delta_t \qquad \forall p, s \tag{10}$$

$$x_{c,s,l},\ y_{t,c,s,l} \in \{0,1\} \qquad \forall t, c, s, l \tag{11}$$

5.2. Modeling Function Requirements

In this section, the modeling of technological requirements is presented in detail, which is captured via function requirements of tasks and components, as well as functions provided by components. These are encoded in the form of constraints in the MILP model of the system configuration problem.

Constraint (12) states that if a function is not provided in a node of the resource hierarchy, then the function requirement is propagated upwards. Here, symbol $\Rightarrow$ denotes logical inference, which can be encoded directly using an indicator constraint in state-of-the-art MILP solvers, or using big-M constraints in less advanced solvers. Inequality (13) states that the amount of function requirement propagated upwards from the line level is at most the amount supplied externally. Function requirements are computed separately for maxitive and additive functions. For maxitive functions (including symbolic functions), the requirement at a node is computed as the maximum of the requirements of components (14), tasks (15), and the requirements propagated upwards from child nodes (16). For additive functions, the underlying idea is similar, but the requirements arising from different components, tasks, and child nodes must be summed up, which is expressed by constraint (17). Functions can be provided only by components that are installed (18). If a component provides a function in a given node, then it must cover the entire amount required, which is limited by the capabilities of the component (19). Equality (20) states that a function is provided in a node if it is provided by a component installed at the node. Constraints (21)-(23) state that functions and energy carriers combined into a function group must be provided by the same component. Binary variables related to function requirements are listed in line (24).

$$(\varphi^P_{f,s,l} = 0) \Rightarrow \ \varphi^U_{f,s,l} \geq \varphi^R_{f,s,l} \qquad \forall f, s, l \tag{12}$$

$$\varphi^U_{f,0,0} \ \leq \ q^X_f \qquad \forall f \tag{13}$$

$$\varphi^R_{f,s,l} \ \geq \ r^C_{f,c}\, x_{c,s,l} \qquad \forall f \in \mathcal{FM}, c, s, l \tag{14}$$

$$\varphi^R_{f,s(t),l(t)} \ \geq \ r^T_{f,t} \qquad \forall f \in \mathcal{FM}, t \tag{15}$$

$$\varphi^R_{f,s,l} \ \geq \ \varphi^U_{f,s',l'} \qquad \forall f \in \mathcal{FM}, s, l, (s', l') \in \mathcal{C}(s, l) \tag{16}$$

$$\varphi^R_{f,s,l} \;=\; \sum_c r^C_{f,c}\, x_{c,s,l} \;+ \sum_{t:\; s=s(t)\;\wedge\; l=l(t)} r^T_{f,t} \;+ \sum_{(s',l')\in\mathcal{C}(s,l)} \varphi^U_{f,s',l'} \qquad \forall f \in \mathcal{AM}, s, l \tag{17}$$

$$\varphi^C_{f,s,l,c} \;\le\; x_{c,s,l} \qquad \forall f, s, l, c \tag{18}$$

$$(\varphi^C_{f,s,l,c} = 1) \Rightarrow \;\; \left(\varphi^R_{f,s,l} \le q^C_{f,c}\right) \qquad \forall f, s, l, c \tag{19}$$

$$\varphi^P_{f,s,l} \;\le\; \sum_c \varphi^C_{f,s,l,c} \qquad \forall f, s, l \tag{20}$$

$$\varphi^C_{f_1,s,l,c} \;=\; \varphi^C_{f_2,s,l,c} \qquad \forall s, l, c, (f_1, f_2) \in \mathcal{G}^{FF} \tag{21}$$

$$\sigma^C_{e_1,s,l,c} \;=\; \sigma^C_{e_2,s,l,c} \qquad \forall s, l, c, (e_1, e_2) \in \mathcal{G}^{EE} \tag{22}$$

$$\varphi^C_{f,s,l,c} \;=\; \sigma^C_{e,s,l,c} \qquad \forall s, l, c, (f, e) \in \mathcal{G}^{FE} \tag{23}$$

$$\varphi^P_{f,s,l}, \varphi^C_{f,s,l,c} \in \{0,1\} \qquad \forall f, s, l, c \tag{24}$$

5.3. Modeling Energy Consumption

This section completes the MILP representation with the formulation of the energy consumption model. Analogously to the handling of function requirements, constraint (25) states that if some energy requirement is not satisfied by generation in a given node of the resource hierarchy, then the energy requirement is propagated upwards. Constraint (26) prohibits propagation upwards from the line level for all energy carriers that are not supplied externally. The indicator constraint (27) computes idle consumption of each component in the configuration. The total energy consumption in a node is calculated from the working-state consumption of task-component pairs, the idle-state consumption of components, the energy converted into another carrier, and the consumption propagated upwards from child nodes (28). In order to simplify the formulas, constraint (27) calculates idle consumption for the entire cycle time, i.e., also for the time spent in working state; in exchange, constraint (28) adds only the difference between working-state and idle consumption, $p^W_{e,c,t} - p^I_{e,c}$, for the time spent in working state. Energy carriers can be provided by generation only by components that are installed

(29) and can generate the given carrier (30). If component c provides energy carrier $E_{\text{to}}(c)$ by generation from $E_{\text{from}}(c)$, then the energy consumption of $E_{\text{from}}(c)$ can be computed from the component's transfer function and the requirement of carrier $E_{\text{to}}(c)$ (31). Equality (32) states that if an energy carrier is provided by generation in a node, then the entire amount must be provided by one converter component. Finally, binary variables used for modeling energy consumption are listed in constraint (33).

$$(\sigma^P_{e,s,l} = 0) \Rightarrow \left(\sigma^U_{e,s,l} \geq \sigma^R_{e,s,l}\right) \qquad \forall e, s, l \tag{25}$$

$$\sigma^U_{e,0,0} = 0 \qquad \forall e \notin \mathcal{EX} \tag{26}$$

$$(x_{c,s,l} = 1) \Rightarrow \left(\sigma^I_{e,s,l,c} = \sum_p p^I_{e,c} V_p \tau_p\right) \qquad \forall e, s, l, c \tag{27}$$

$$\sigma^R_{e,s,l} = \sum_{t,c} \left(p^W_{e,c,t} - p^I_{e,c}\right) d_{t,c} V_{p(t)} y_{t,c,s,l} + \sum_c \sigma^I_{e,s,l,c} + \sum_c \sigma^G_{e,s,l,c} + \sum_{(s',l') \in \mathcal{C}(s,l)} \sigma^U_{e,s',l'} \qquad \forall e, s, l \tag{28}$$

$$\sigma^C_{e,s,l,c} \leq x_{c,s,l} \qquad \forall e, s, l, c \tag{29}$$

$$\sigma^C_{e,s,l,c} = 0 \qquad \forall e, s, l, c : e \neq E_{\text{to}}(c) \tag{30}$$

$$(\sigma^C_{E_{\text{to}}(c),s,l,c} = 1) \Rightarrow \left(\sigma^G_{E_{\text{from}}(c),s,l,c} = a_c \sigma^R_{E_{\text{to}}(c),s,l} + b_c \sum_p V_p \tau_p\right) \qquad \forall s, l, c \tag{31}$$

$$\sigma^P_{f,s,l} \leq \sum_c \sigma^C_{f,s,l,c} \qquad \forall e, s, l \tag{32}$$

$$\sigma^P_{e,s,l}, \sigma^C_{e,s,l,c} \in \{0, 1\} \qquad \forall e, s, l, c \tag{33}$$

5.4. Overall Mathematical Model

With the above definitions, the overall mathematical model of the energy-aware system configuration problem is formulated as a MILP, solvable efficiently by state-of-the-art mathematical solvers, as follows:

minimize

$$w_I \sum_{c,s,l} k_c^C x_{c,s,l} \; + \; w_E \sum_{e} k_e^E \sigma_{e,0,0}^U \tag{34}$$

subject to

$$(2) - (11), (12) - (24), (25) - (33). \tag{35}$$

6. Decision-support Tool

The proposed mathematical models and solution approaches were implemented in a web-based decision-support tool comprising two main modules: *System Configurator* for automatically computing or manually editing configurations, and *Energy Evaluator* for performing detailed simulation-based energy assessments of the configurations. The following subsections describe the functionality of each module.

6.1. Overview of Functionalities: System Configurator

The System Configurator module of the decision-support tool facilitates the energy-aware configuration of production systems via the following main functionalities:

- The software maintains a resource library, containing the applicable components, their properties, the functions they provide and require, as well as their energy consumption characteristics, see Figure 8. These models are common over all configuration projects in the software tool.
- It allows the definition and maintenance of different projects, each targeting the optimal configuration of a production system for a given set of requirements. These requirements, including the description of the products, their process plans, as well as all parameters of the mathematical model, e.g., the cycle time limits and the objective function weights, must be provided by the user.
- Based on the above component library and project data, it automatically computes the optimal production system configuration using the solution approach presented earlier in Section 5. The configuration is displayed on the GUI in two different views, a components view and a task assignments view, accompanied by a brief evaluation including energy consumption, energy and investment costs, as well as actual cycle times, see Figure 9.

- Finally, the software supports interactive configuration processes by enabling the iterative combination of manual editing operations and automated computation steps. The user can freely override constraints or decisions of the formal mathematical approach via the following manual modifications: (1) fixing a specific component in a given position of the resource hierarchy; (2) assigning a specific component for a task; or (3) filtering the applicable components based on one or multiple component properties for the overall project or for specific tasks. In case the manual modifications are inconsistent with the formal model, e.g., the component manually assigned to a task does not meet its function requirements, the software issues a warning message. Yet, user decisions always dominate those proposed by the automated solver.

  This possibility of manual editing, mixed-initiative planning, as well as explainable optimization is particularly important in practical engineering optimization where the discrepancies between the implemented mathematical model and the user's understanding of the real-world problem are inevitable. Early results on explainable optimization in production system configuration have been presented in [40]. The complete integration of the explainable optimization methods proposed there into the decision-support tool is a direction for future work.

6.2. Overview of Functionalities: Energy Evaluator

To enable energy-focused decision making, the Energy Evaluator complements the System Configurator by the detailed assessment of the energy performance of a candidate system configuration. It provides simulation-based feedback on the energetic and environmental implications of configuration choices, thereby supporting the selection of energy-efficient system designs. It provides the following main functionalities:

- It performs simulation-based analyses to generate configuration-specific load profiles by deriving the detailed schedule of tasks and corresponding component energy states over time.

- It computes the following energy-related key performance indicators (KPIs), derived from the simulated load profiles and the corresponding evaluation models: total energy consumption, energy consumption per

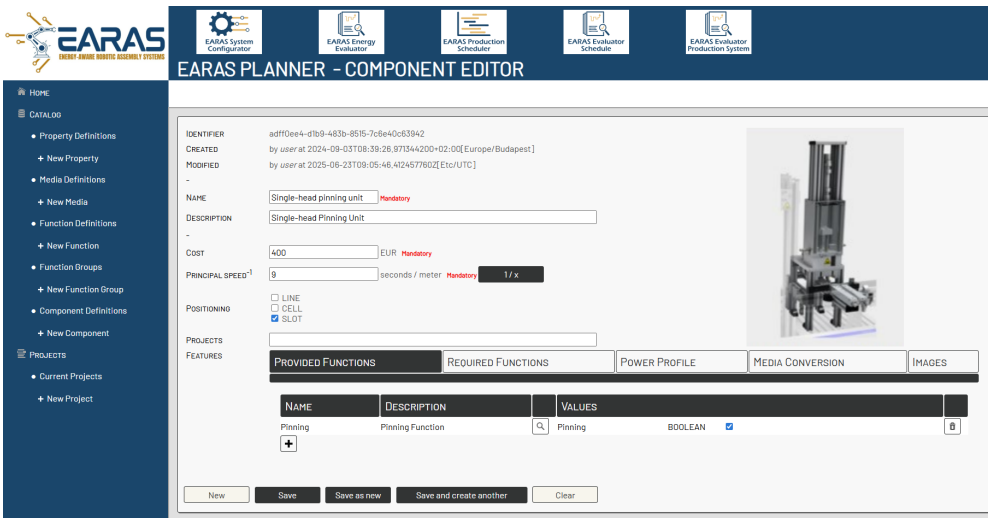

Figure 8: Screenshot from the decision-support tool: specification of a component in the component library.

product and per cycle, value-adding (working-state) and non-value-adding (idle) energy consumption, and peak load. In addition, energy costs and total costs are calculated for the system as a whole, per product, and per cycle. Unlike System Configurator, the calculation of KPIs takes into account the timing and occurrence of the values, so that the ramp-up and ramp-down of production are factored in. Especially in the case of slow-moving, energy-intensive, or small-batch production systems, these energy consumption figures can contribute significantly to total consumption.

- Finally, the results are presented interactively via a graphical user interface, including time-resolved load profiles, KPI summaries (see Figure 10), and detailed task sequences in the form of Gantt charts, enabling users to analyze system behavior and identify energy optimization potentials.

For the evaluation of the configuration in economic and ecological dimensions, the total costs (C) as well as the amount of greenhouse gas emissions (E) expressed as the carbon footprint in the unit of carbon dioxide equivalents (kg_{CO_2e}) were selected. The cost formula is based on [45] but was simplified to fit the applied energy consumption model:

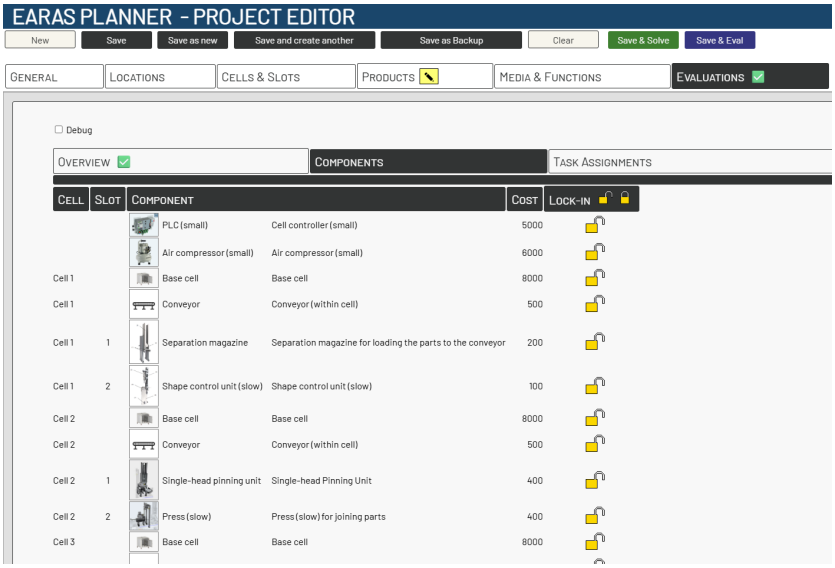

Figure 9: Screenshot from the decision-support tool: components view of the optimal system configuration for a project.

$$C = C_W + C_E$$

Resource costs (costs that are related to the initial investment, operation, and maintenance of resources, C_W) are calculated by multiplying occupancy times with the respective resource cost factors. The overall resource cost is the sum of the individual component costs. Energy costs (C_E) are composed of line-induced grid energy costs, grid peak load costs, self-generation energy costs and revenues from grid feed-in. For reasons of simplification and because of the assembly line focus of the approach, non-line-induced shares of these cost types (e.g., from auxiliary or building-related energy users) were not considered in the evaluation.

6.3. Software Implementation

The System Configurator is implemented as a single HTML5-based Java Enterprise Edition web application. Graphical outputs are rendered using embedded SVG images. All data are stored and transmitted in JSON format. The data structures are documented and illustrated with examples in the

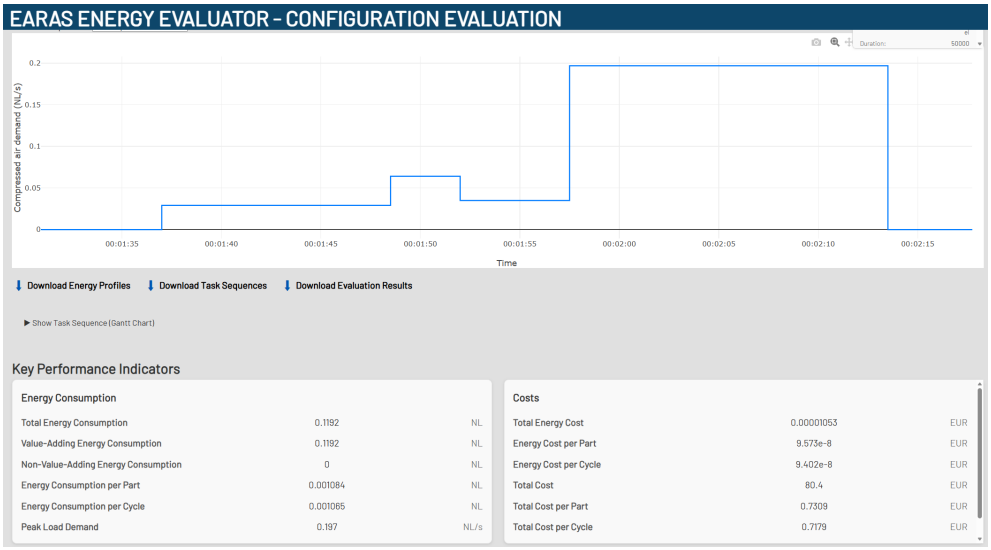

Figure 10: Screenshot from the decision-support tool: display of the results of a configuration evaluation.

online repository[1]. The production system configuration problem is solved using the FICO Xpress 9.4 mathematical optimization solver.

The energy evaluator is implemented as a Python-based web application, combining a simulation-driven backend with a browser-based front end. The backend uses Flask to provide HTTP interfaces and orchestrate discrete-event simulations implemented with SimPy. Input and output data are consistently stored and transmitted in JSON format, enabling integration with external components. Interactive graphical outputs, such as load profiles and evaluation results, are rendered in an HTML5/CSS3 and Vanilla JavaScript interface. The underlying data structures and exemplary datasets are also documented and provided in the online repository.

7. Validation

A fourfold validation of the proposed methodology has been performed. First, the practical applicability of the proposed modeling framework is demonstrated by encoding the mMS sample use case. Second, the obtained

[1]https://git.sztaki.hu/emi/earas-articles

configuration is analyzed through Pareto and sensitivity analyses. Third, the prediction accuracy of the method regarding energy consumption is demonstrated via three product variants in the mMS use case. Finally, experiments are conducted on randomly generated large-size problem instances to investigate the computational efficiency of the approach.

7.1. Modeling the Sample Use Case

The applicability and soundness of the proposed modeling framework were validated by encoding the mMS sample use case, presented earlier in Section 3.1. Modeling the existing physical process first required identifying the production tasks and the corresponding components in such a way that the tasks capture continuous, non-overlapping segments of the process, and each task can be characterized by clearly defined start and end times, as well as by a single component that executes it. This resulted in 17 tasks and 6 corresponding primary components, as shown in Table A.4 in the Appendix. Moreover, three important auxiliary components providing different services to the primary components were also added to the model: (1) the base cell, which contains the common mechanical, electrical, and pneumatic infrastructure for the primary components; (2) the PLC, which controls the operation of a single station or of the entire system; and (3) the air compressor, which provides compressed air to pneumatic components.

The technological requirements of the tasks and the components were captured by 10 *functions*. This includes 7 *symbolic* functions encoding the capabilities of the 6 primary components. In most cases, the capabilities of a component are encoded in a single function; however, the press executes two types of tasks: it first loads the lower part (*press loading* function), and then loads the upper part and joins the two parts (*pressing* function). Since these two functions must be provided by the same press component, they are grouped together into a *function group*.

Regarding the functions provided by auxiliary components to primaries, controlling the operation of different components is captured by the *PLC channels* additive function: the PLC must provide as many I/O channels as the *sum* of the channel requirements of the controlled components. *Compressed air pressure* is a maxitive function: all individual components can be served by a single compressor that provides the maximum of the required pressure values.

Finally, two energy carriers are considered: *electricity*, with consumption measured in watt-hours (Wh) and *compressed air*, measured in normal liters

(Nl). Since the compressed air energy carrier and the air pressure maxitive function must be provided by the same compressor, these two are combined into a function group.

Regarding energy consumption, tasks executed by pneumatic components are characterized by their working-state compressed air consumption. The duration of the working-state energy block is equal to the duration of the task. No idle consumption of compressed air is modeled, which corresponds to the assumption that there is no leakage in the pneumatic system.

Electricity consumption measurements performed on the AC and DC sides of the physical mMS system showed that, somewhat surprisingly, the final AC consumption of the system is practically independent of the executed tasks. This is due to the low DC-side power demand and the inefficiency of AC/DC conversion. Accordingly, electricity consumption was captured as the idle consumption of the base cells.

Additional electricity consumption occurs at the air compressor in the form of indirect consumption caused by the hysteresis control of the compressed air tank. This consumption was characterized by the transfer function of the air compressor, identified from repeated measurements involving the controlled emptying and refilling of the tank.

The process plan of the basic product, including tasks, durations, and compressed air consumption is detailed in Table A.4 in the Appendix. An overview of these tasks and their temporal relations were presented earlier in Figure 2.

7.2. Solving the Configuration Problem: Pareto and Sensitivity Analyses

A Pareto analysis of the two objectives, investment and energy costs, together with a sensitivity analysis of the cycle time limit are conducted to demonstrate the practical value of the proposed solution approach for decision support. In these experiments, the component library includes all components present in the physical mMS system, with true measured task duration and energy consumption data; as well as alternative components to illustrate different features of the proposed approach. Consequently, for almost all component types in the mMS use case, alternative choices are available, allowing selection between a slower, smaller, and less expensive component and a faster, larger, and more expensive one.

The configuration obtained by minimizing only the investment cost, with a sufficiently large cycle time limit, is considered to be the *baseline configuration*. For this solution, the slower, lower-cost components are selected in

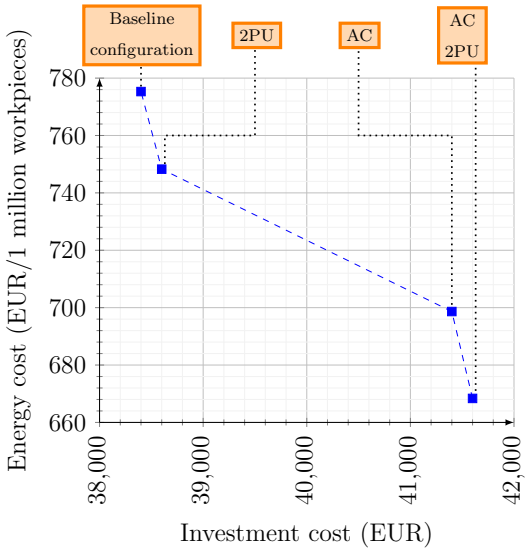

Figure 11: Pareto frontier between the investment and energy costs in the sample use case. The labels above the diagram indicate the changed components compared to the baseline configuration, together with their station number. 2PU stands for pinning unit in Station 2, while AC for air compressor.

all cases, resulting in an investment cost of 38 400 EUR, an energy cost of 775 EUR, and a cycle time of 44.5 seconds.

Figure 11 shows the Pareto frontier between the two elements of the objective function, whereas Figure 12 reports the optimal configurations and their costs for different cycle time limits under equal objective function weights. Both figures compare the corresponding configurations with the baseline case. The Pareto frontier consists of four configurations, and the cycle time sensitivity analysis identifies five relevant solutions, with only two appearing in both sets.

The configurations comprising the Pareto frontier differ in the selected pinning unit in Station 2 and the installed air compressor. The most energy-efficient configuration is associated with 668 EUR energy cost, 41 600 EUR investment cost, and a cycle time of 44.5 seconds. From the 107 EUR reduction of energy costs compared to the baseline configuration, roughly 37 EUR comes from the decreased cycle time and the corresponding lower idle electricity consumption, made possible by the more efficient pinning unit. Conversely, the same pinning unit has a higher compressed air consumption, increasing the energy costs by 10 EUR. Finally, the more efficient air com-

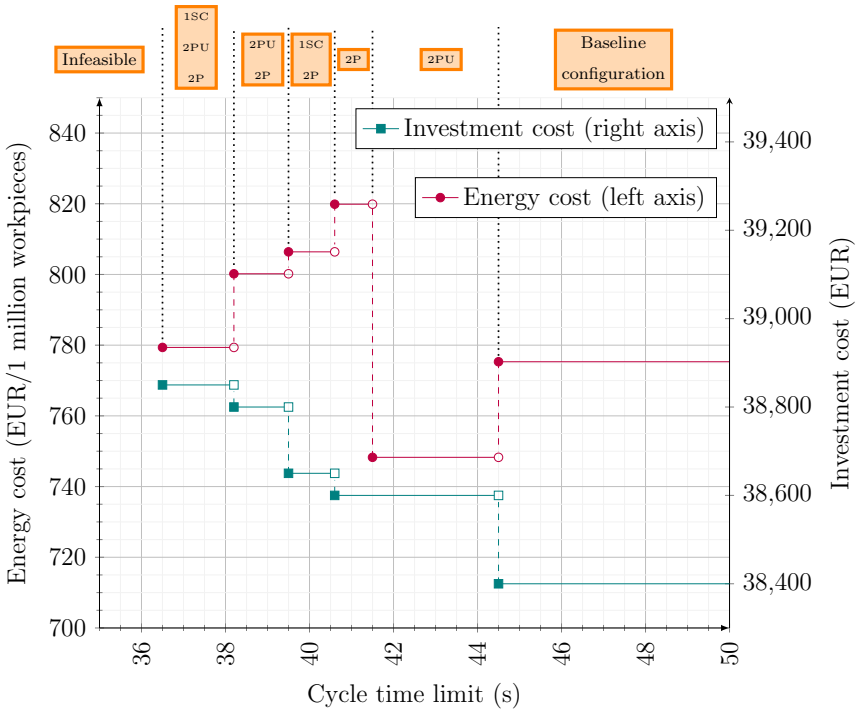

Figure 12: Investment and energy costs of the optimal solutions with equal objective function weights and different cycle time limits. The labels above the diagram indicate the changed components compared to the baseline configuration, together with their station number. 2PU stands for pinning unit in Station 2, 2P for press in Station 2, 1SC for shape control unit in Station 1.

pressor lowers the energy costs by 80 EUR. In addition to illustrating the trade-off between the different objectives, changes in energy prices would also lead to one of the configurations along the Pareto frontier being optimal.

The cycle time sensitivity analysis presented in Figure 12 shows that the baseline configuration has the highest cycle time of 44.5 seconds among the relevant solutions, whereas the fastest possible line has a cycle time of 36.5 seconds. As the cycle time limit decreases (the relevant constraint becomes stricter), the total cost increases. At the same time—as the changes in the configurations suggest—the bottleneck station alters repeatedly between the first and second stations. However, for some component types, there are several switches between the slower and faster units, some of which may initially seem counterintuitive, see, e.g., the shape control unit in the first station. This pattern arises from the synchronization constraints, which significantly influence station cycle time calculations. Some components affect the cycle times of multiple stations through all of their assigned tasks (e.g., the sole task assigned to the pinning unit is involved in the synchronization of Stations 1 and 2) or through only a subset of their assigned tasks (e.g., one of the two tasks of the press is member of the synchronized task sets), see Figure 2.

The results demonstrate the value of the proposed approach as a decision-support tool by identifying different optimal configurations under varying objectives and constraints faced by the decision maker. They also show that small changes in the input parameters lead to correspondingly moderate changes in the resulting configurations.

7.3. Verification of the Energy Consumption Model

In order to verify the proposed energy consumption model, three product variants in the mMS use case were considered with well-distinguishable complexity, and consequently consumption levels (high, medium, and low). The low-complexity product variant was a version of the product described in Section 7.1. Medium- and high-complexity product variants were created by modifying the process parameters of several tasks, as shown in Table A.5 in the Appendix.

The process of validation for the prediction accuracy of the proposed method consisted of the following steps:

1. Energy consumption and task duration data were collected via measurements about the production of a single workpiece for each of the

three product variants. Measured data included the duration of each task (seconds); current (amps) measured by an ATO-CUS-AC1500 Hall sensor; and compressed air flow (normal liters per minute) measured by an IFM SD5000 sensor. Consumption was measured for the whole line at each tenth of a second during production.

2. Based on this information, a baseline ES model was set up for the high-complexity product variant.

3. Applying the real (measured) task duration values, and the energy intensity parameters calibrated to the high-complexity variant, the energy consumption of the medium- and low-complexity variants were predicted.

4. The accuracy of the ES model was assessed by comparing the predicted and measured consumption values.

The process above emulates the practical problem where energy consumption measurements are available for an existing product variant, whereas predictions are required for similar, future variants for the design of a new or updated manufacturing system.

As measured electricity consumption was practically constant, its average value was assigned to the base cells as idle consumption. Conversely, compressed air consumption showed substantial variation over time, while no leakage was detected. Accordingly, air consumption was modeled as the working-state consumption of the executed tasks. In order to mitigate measurement errors that resulted in small air consumption values occurring after (and rarely, also before) executing pneumatic tasks, these values were reassigned to the nearest pneumatic task in time. As compressed air consumption varies significantly even during the execution of a task, the modification of process parameters presented in Table A.5 could potentially result in inaccurate predictions.

	Electricity			Compressed air		
Product variant	Predicted	Measured	Error	Predicted	Measured	Error
High-compl.	7.2972 Wh	7.2972 Wh	0.00%	8.4617 Nl	8.4617 Nl	0.00%
Medium-compl.	6.3178 Wh	6.3262 Wh	0.13%	6.7477 Nl	6.7385 Nl	0.14%
Low-compl.	4.7730 Wh	4.7669 Wh	0.13%	4.5276 Nl	4.5203 Nl	0.16%

Table 3: Results of the energy consumption prediction accuracy verification.

The results of the prediction accuracy test are presented in Table 3. As expected, prediction errors are zero for the high-complexity product variant, which is used for model calibration. For the other two variants, the errors remain below 0.2%, both for electricity and compressed air. It should be emphasized that this extremely high accuracy was achieved because all processes in the mMS system are highly deterministic, and because of the relatively small difference between the existing system, represented by the measured high-complexity variant, and the future system, represented by the medium- and low-complexity target variants. In many industrial applications, systems may be subject to substantially greater uncertainty, or the differences between the existing and future systems may be more pronounced.

In real-world manufacturing lines, predictions based on discrete energy blocks exhibit greater variability. For instance, in a comprehensive study on automotive body shop lines across over 200 workgroups, Teiwes et al. [14] reported prediction errors between 15% and 30% for energy block models. Crucially, as highlighted by [14], traditional early-stage design typically operates with over-dimensioning and estimation uncertainties of 300% to 400%. Consequently, even with industrial error margins of 15–30%, the underlying energy block modeling approach provides substantial accuracy improvements for early-stage system configuration decisions.

7.4. Computational Evaluation

While the solution of the above variants of the small-scale sample problem took only 0.1-0.2 seconds, computational efficiency may become a relevant aspect for large-scale industrial problems. For this reason, computational experiments investigated the efficiency of the proposed solution approach. Preliminary experiments identified problem size, objective function, and the external supply of energy carriers as factors that substantially affect solver performance. In contrast, other parameters, such as the tightness of the cycle time limits, were found to have very little impact.

Benchmark problem instances were generated based on the mMS use case, by preserving the problem structure (types of tasks and components, function requirements, energy carriers, etc.), but building a set of large, diverse, and computationally challenging instances. Instance size was increased by duplicating the stations (from the original 3 stations to $2 \times 3 = 6$, $4 \times 3 = 12$, or $6 \times 3 = 18$ stations), tasks (from 2-8, on average 5.67 tasks per station in the original mMS use case to $2 \times 5.67 = 11.33$, $4 \times 5.67 = 22.67$ or $6 \times 5.67 = 34$

tasks per station on average in the instances), as well as the number of component types (from the original 9 component types to $2 \times 9 = 18$, $4 \times 9 = 36$, or $6 \times 9 = 54$ component types). Numeric parameters, including component costs, task durations, working-state and idle energy consumption were randomized by a $\pm 50\%$ perturbation of the original mMS parameter values. Cycle time limits were set according to the average durations over the different execution modes of the tasks at each station, i.e., roughly the half of the combinations of execution modes satisfy the cycle time limit.

Three different objective functions were considered: investment cost ($w_I = 1$, $w_E = 0$), energy cost ($w_I = 0$, $w_E = 1$), and total cost ($w_I = w_E = 1$). All instances were solved with and without compressed air supplied externally. In the latter case, the solver must select the most appropriate air compressor from the component library.

The above generation scheme, comprising 3^3 different instance sizes, 3 objective functions, and 2 options for the availability of compressed air, resulted in $3^3 \times 3 \times 2 = 162$ instance classes. Generating 5 random instances for each class yielded a total of $162 \times 5 = 810$ instances.

A standalone solver implemented in FICO Xpress v9.4 was used for the experiments, which included the same MILP model as embedded in the web-based decision-support tool. Experiments were run on a personal computer with Intel i7 1.80 GHz CPU and 16 GB RAM, with a time limit of 600 seconds per instance.

A summary of the computational results is displayed in Figures 13 and 14, whereas detailed results are presented in Tables B.6-B.11 in the Appendix. The results indicate that very large problem instances, namely, over 90% of the instances with up to 12 stations, 204 tasks and 54 component types, could be solved to proven optimality. For the remaining instances, the solver hit the time limit, and terminated with a valid but possibly sub-optimal solution.

Yet, performance depends heavily on instance characteristics, as displayed in Figure 13: minimizing investment cost (97.4% of the instances solved to optimality, 18.8 s average computation time, 0.3% average gap) is significantly easier than minimizing total cost (72.2% optimal, 176.3 s computation time, 2.7% gap), whereas the most challenging objective is clearly minimizing energy cost alone (48.8% optimal, 316.9 s computation time, 12.2% gap). The reason is that investment cost is expressed simply as a weighted sum over the core decision variables $x_{c,s,l}$ in the objective (1) of the MILP, whereas energy cost is inferred via a complex network of auxiliary binary variables and

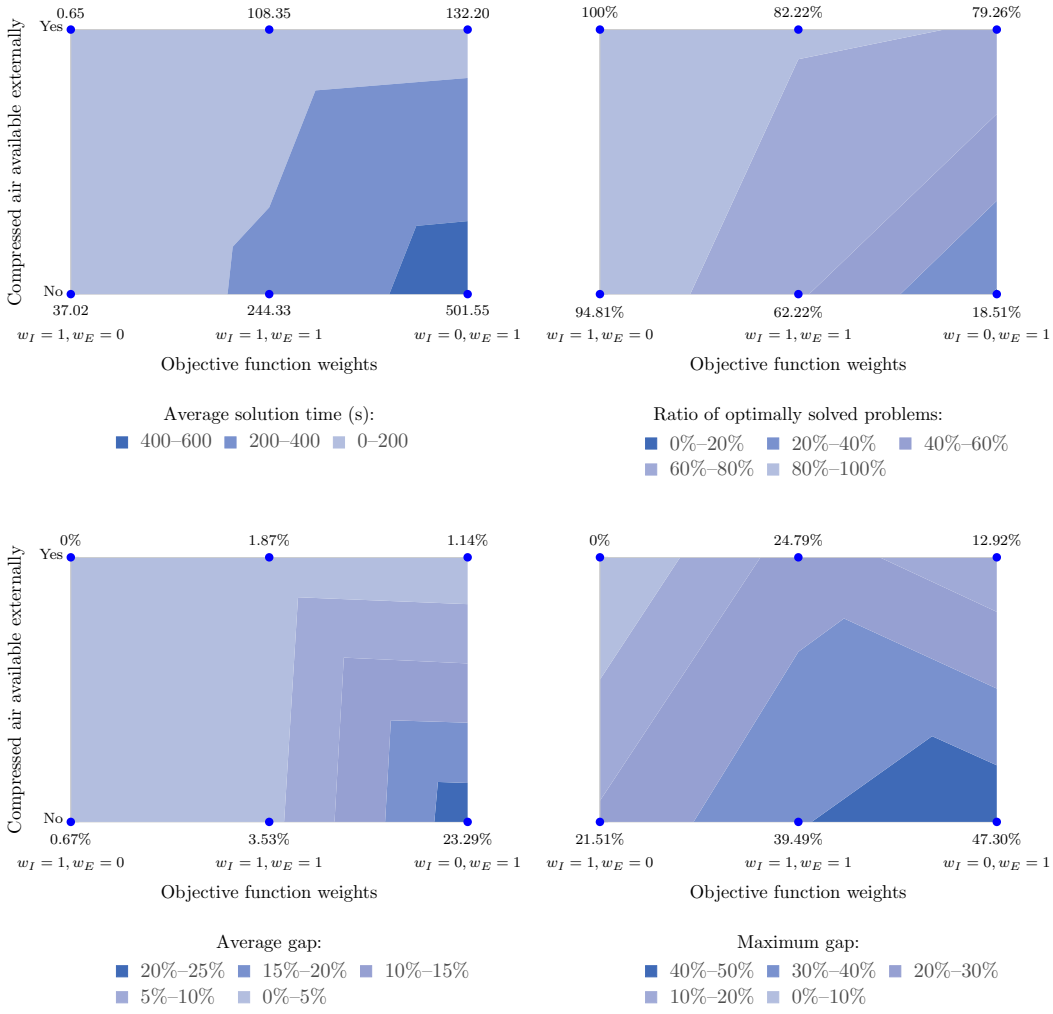

Figure 13: Computational results by instance characteristics: heat maps displaying the average solution time (top left), ratio of problem instances solved to optimality (top right), average gap (bottom left), and maximum gap (bottom right) depending on objective function weights (horizontal axis) and compressed air supply (vertical axis).

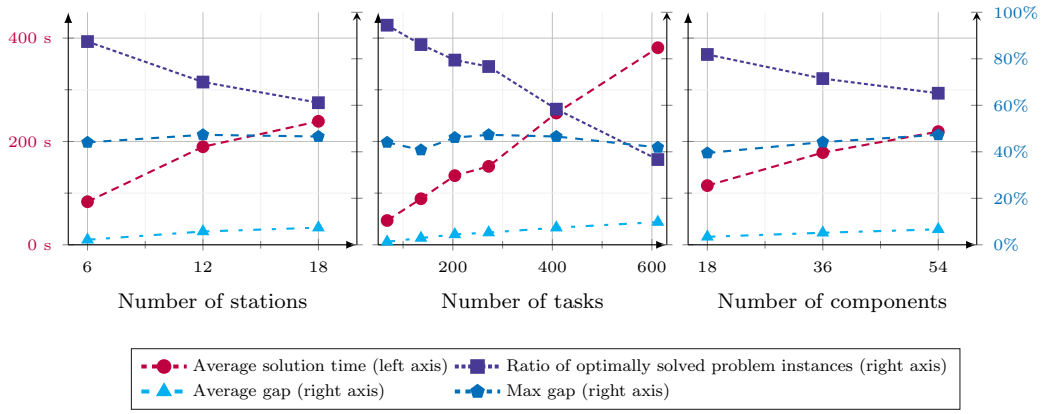

Figure 14: Computational results by instance size: number of stations (left), tasks (middle) and components (right).

constraints, including indicator (logical inference) constraints. Total cost is easier than energy cost alone because the solver can use the easy-to-compute investment cost component as a strong lower bound. Nevertheless, it should be noted that energy cost as a single objective is hardly relevant in practical applications.

Moreover, instances where compressed air is supplied externally (87.2% optimal, 80.4 s computation time, 1.0% gap) are easier to solve than instances where an air compressor must be installed (58.5% optimal, 261.0 s computation time, 9.2% gap). Like above, the reason is the mathematically complex model of energy conversion, using binary variables and indicator constraints. Finally, as expected, performance degrades with the increase of instance size, which includes the decrease of the ratio of instances solved to proven optimality, as well as the increase of computation time and optimality gaps, see Figure 13.

The average optimality gap was 5.1% considering all instances, and 18.7% over the instances where the time limit was hit. This includes large gaps on a few outlier instances: the largest optimality gap encountered was 47.3%. It is noted that the rapid increase of the optimality gap depending on instance size and characteristic is a common phenomenon when solving large-scale MILPs using exact solvers subject to a fixed time limit. The reported gaps reflect partly the sub-optimality of the computed solution, and partly the looseness of the lower bound. The relative contribution of these two factors can be determined only when the exact optimal solution is known.

Overall, these results suggest that the approach is readily applicable to industrially relevant problem sizes. In extreme environments where even larger instances must be tackled, the development of efficient heuristics can be an interesting direction for future research.

8. Conclusions and Future Research

8.1. Research Implications

This paper proposed a novel approach to the energy-aware configuration of production systems. The approach relies on the following key elements: (1) an energy state model of the production system, characterizing energy consumption at the level of individual system components and tasks; (2) the integration of the energy state model into a formal mathematical model of the production system configuration problem; and, as a result, (3) an overall decision-support framework that enables the consideration of the energy consumption of production systems already in the early design stage, and therefore, effectively minimizing the total investment and energy cost of the future system.

In this way, the proposed approach supports the energy-aware configuration of production systems, and substantially extends classical configuration methods, which typically consider only investment cost during system design and evaluate energy efficiency only a posteriori.

A case study on a physical production line confirmed the applicability of the proposed approach to real industrial scenarios. In particular, the energy state model enabled highly accurate prediction of electricity and compressed air consumption when extrapolating from measured data for an existing product variant to a future variant with a different process plan. This accuracy is partly attributable to the highly deterministic and relatively simple production environment, which should be considered when comparing these results with those reported for more complex industrial environments.

An important limitation of the proposed approach is that, among various practically relevant energy- and sustainability-related objectives, it focuses exclusively on minimizing energy consumption or cost, assuming a known and fixed (or average) unit energy cost for large-batch production. A richer set of objectives, including line-level peak load, self-generation, carbon footprint, and ramp-up/ramp-down events in small-batch production, is evaluated a posteriori by the Energy Evaluator module. Some of these objectives can also be integrated into the optimization model. For example, finite batch

sizes and the resulting ramp-up/ramp-down activities can be captured by incorporating these activities into the energy state model and assuming a representative batch size. Others require substantially extending the model scope. Line-level peak load, for instance, requires time-varying task consumption in the energy state model and detailed task scheduling. Plant-level peak load, in turn, requires extending the scope to the entire plant.

The most relevant direction for future research is the development of reliable methods for constructing energy state models for novel products and processes, especially in cases where these differ significantly from previously measured variants. In addition, extending the approach toward the explicit integration of energy flexibility into production system configuration is of particular interest. This includes the consideration of technologies such as variable-speed drives, flexible heating and cooling systems, and multi-energy equipment, enabling the adaptive operation of production systems in response to the current energy environment.

8.2. Implications for practitioners

Efforts to enforce energy efficiency during the early-stage design of production systems are hampered by the limited availability of detailed knowledge about the energy consumption characteristics of system components and manufacturing tasks. In many applications, only high-level measurements (typically at the plant level, or at best at the production line level) are available. Consequently, developing a detailed understanding of energy consumption behavior is of utmost importance for improving the energy efficiency of production systems.

To address this issue, an energy state model was proposed that characterizes energy consumption at the level of individual tasks and components. It was demonstrated that the proposed model can be directly embedded into formal mathematical models that are tractable by off-the-shelf optimization solvers. This enables the effective estimation and minimization of energy costs already during the early stages of manufacturing system design. These techniques were implemented in a prototype decision-support software tool that facilitates a mixed-initiative configuration process, combining automated problem solving with manual modifications to accommodate specific user preferences.

When the target system differs only slightly from an existing system for which measured data are available, the energy state model provides a surprisingly accurate estimate of future energy consumption. In cases where

the differences are significant, additional research is required; in this context, the application of machine learning techniques appears to be a promising direction.

In the case of the physical production line investigated in the experiments, several measured characteristics were unexpected even for experienced engineers. Due to the inefficiency of AC/DC conversion, the activity of DC-powered components had no observable impact on the overall AC power consumption. As a result, total energy consumption could only be reduced by decreasing the cycle time or by improving the efficiency of AC/DC conversion. In contrast, compressed air consumption and pneumatic components have a direct impact on energy costs. First, potential air leakages must be eliminated; furthermore, investment into more energy efficient pneumatic components can be economically justified. Similar easy-to-follow rules of thumb can be highly valuable for practitioners; however, establishing generic and reliable guidelines for system design and operation requires insights gained from a large and diverse set of applications. Developing such guidelines is therefore an important direction for future research.

Acknowledgments

The authors would like to thank their colleagues in the EARAS consortium from Hepenix and Blue Yonder for their valuable support, particularly for granting access to their physical mMS line and for providing the energy consumption data for validation. The authors gratefully acknowledge the financial support provided by the German Federal Ministry of Research, Technology and Space (project ID: 02P22D001) and the National Research, Development and Innovation Office, Hungary (project ID: 2020-1.2.3-EUREKA-2022-00014), within the framework of the EUREKA project 2021-17376/NP/BILAT_HU_DE_1. The authors also acknowledge the administrative support provided by Projektträger Karlsruhe.

CRediT authorship contribution statement

András Kovács: conceptualization, methodology, validation, writing–original draft. **Zsombor Szádoczki:** conceptualization, methodology, validation, writing–original draft. **Dávid Karnok:** methodology, software. **Lukas Wittmann:** conceptualization, methodology, software, validation, writing–original draft. **Vincent Kalchschmid:** methodology, validation, writing–original draft.

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.

References

[1] European Commission, Energy prices and costs in Europe, Tech. Rep. COM(2025) 72 final, European Commission (2025).
URL https://energy.ec.europa.eu/data-and-analysis/energy-prices-and-costs-europe_en

[2] C. U. Akpuokwe, A. O. Adeniyi, S. S. Bakare, N. E. Eneh, Legislative responses to climate change: A global review of policies and their effectiveness, International Journal of Applied Research in Social Sciences 6 (2024) 225239.

[3] J. Poligkeit, T. Fugger, C. Herrmann, Decarbonization in the automotive sector: A holistic status quo analysis of original equipment manufacturer strategies and carbon management activities, Sustainability 15 (2023) 15753.

[4] T. AlGeddawy, H. ElMaraghy, Design for energy sustainability in manufacturing systems, CIRP Annals 65 (1) (2016) 409–412.

[5] S. Mainwaring, Why sustainability is critical for business success in 2025 (and the data to prove it), Forbes (2025).
URL https://www.forbes.com/sites/simonmainwaring/2025/01/30/why-sustainability-is-critical-for-business-success-in-2025-and-the-data-to-prove-it/

[6] R. Menghi, A. Papetti, M. Germani, M. Marconi, Energy efficiency of manufacturing systems: A review of energy assessment methods and tools, Journal of Cleaner Production 240 (2019) 118276.

[7] Z. Cao, D. Huang, S. Xu, Minimizing cycle time and energy consumption in robotic assembly line: A multi-objective metaheuristic, Computers & Operations Research 192 (2026) 107501.

[8] N. Weinert, S. Chiotellis, G. Seliger, Methodology for planning and operating energy-efficient production systems, CIRP Annals 60 (1) (2011) 41–44.

[9] A. Mayr, T. Lechler, T. Donhauser, M. Metzner, E. Schäffer, E. Fischer, J. Franke, Advances in energy-related plant simulation by considering load and temperature profiles in discrete event simulation, Procedia CIRP 81 (2019) 1325–1330.

[10] Y. Zhang, L. Li, W. Liu, L. Li, Y. Gao, W. Cai, J. W. Sutherland, Dynamics analysis and energy consumption modelling based on bond graph: Taking the spindle system as an example, Journal of Manufacturing Systems 62 (2022) 539–549.

[11] J. Wójcicki, T. Tolio, G. Bianchi, Cross-level model of a transfer machine energy demand using a two-machine generalized threshold representation, Journal of Manufacturing Systems 58 (2021) 44–58.

[12] Y. Alaouchiche, Y. Ouazene, F. Yalaoui, Economic and energetic performance evaluation of unreliable production lines: An integrated analytical approach, IEEE Access 8 (2020) 185330–185345.

[13] P. C. Priarone, M. C. Magnanini, An approach for integrating performance evaluation and environmental sustainability assessment for hybrid additive-subtractive manufacturing, CIRP Journal of Manufacturing Science and Technology 64 (2026) 65–78.

[14] H. Teiwes, M. Rössinger, M. Gonter, C. Herrmann, S. Thiede, Data based analysis and improvement of energy efficiency in the automotive body shop, Journal of Cleaner Production 284 (2021) 125269.

[15] P. Liu, Z. Zhang, X. Wang, X. Li, X. V. Wang, J. Tuo, A generalized method for the inherent energy performance modeling of machine tools, Journal of Manufacturing Systems 61 (2021) 406–422.

[16] Verband Deutscher Maschinen- und Anlagenbau, VDMA 34179, Messvorschrift zur Bestimmung des Energie- und Medienbedarfs von Werkzeugmaschinen in der Serienfertigung: Measurement instruction to determine the energy- and resource demand of machine tools for mass productions, VDMA-Einheitsblatt, Beuth Verlag GmbH, 2019.

[17] N. Boysen, M. Fliedner, A. Scholl, A classification of assembly line balancing problems, European Journal of Operational Research 183 (2) (2007) 674–693.

[18] A. Scholl, C. Becker, State-of-the-art exact and heuristic solution procedures for simple assembly line balancing, European Journal of Operational Research 168 (3) (2006) 666–693.

[19] N. Boysen, P. Schulze, A. Scholl, Assembly line balancing: What happened in the last fifteen years?, European Journal of Operational Research 301 (3) (2022) 797–814.

[20] P. Chutima, A comprehensive review of robotic assembly line balancing problem, Journal of Intelligent Manufacturing 33 (2022) 1–34.

[21] Y. Kara, C. Özgüven, N. Yalçın, Y. Atasagun, Balancing straight and U-shaped assembly lines with resource dependent task times, International Journal of Production Research 49 (21) (2011) 6387–6405.

[22] T. L. Urban, W.-C. Chiang, Designing energy-efficient serial production lines: The unpaced synchronous line-balancing problem, European Journal of Operational Research 248 (3) (2016) 789–801.

[23] J. Mukund Nilakantan, G. Q. Huang, S. Ponnambalam, An investigation on minimizing cycle time and total energy consumption in robotic assembly line systems, Journal of Cleaner Production 90 (2015) 311–325.

[24] A. Nourmohammadi, T. Arbaoui, M. Fathi, A. Dolgui, Balancing human–robot collaborative assembly lines: A constraint programming approach, Computers & Industrial Engineering 205 (2025) 111154.

[25] Y. Jiao, Y. Wang, X. Su, F. Wang, An ant colony hybrid simulated annealing algorithm for collaborative optimization of robotic mixed-model parallel two-sided assembly lines balancing, Computers & Operations Research 182 (2025) 107113.

[26] Z. Li, Q. Tang, L. Zhang, Minimizing energy consumption and cycle time in two-sided robotic assembly line systems using restarted simulated annealing algorithm, Journal of Cleaner Production 135 (2016) 508–522.

[27] J. M. Nilakantan, Z. Li, Q. Tang, P. Nielsen, Multi-objective co-operative co-evolutionary algorithm for minimizing carbon footprint and maximizing line efficiency in robotic assembly line systems, Journal of Cleaner Production 156 (2017) 124–136.

[28] Z. Zhang, Q. Tang, Z. Li, L. Zhang, Modelling and optimisation of energy-efficient U-shaped robotic assembly line balancing problems, International Journal of Production Research 57 (17) (2019) 5520–5537.

[29] L. Belkharroubi, K. Yahyaoui, Solving the energy-efficient robotic mixed-model assembly line balancing problem using a memory-based cuckoo search algorithm, Engineering Applications of Artificial Intelligence 114 (2022) 105112.

[30] X. Wen, W. Wang, H. Li, S. Ji, W. Jiang, L. Zhang, W. Yang, An improved multi-objective harmony search algorithm based on reinforcement learning for solving automated assembly line balancing and energy consumption optimization problems, Computers & Industrial Engineering 210 (2025) 111509.

[31] O. Yilmaz, N. Aydin, I. Kucukkoc, Bi-objective optimization of human-robot collaborative mixed-model multiple assembly lines considering model assignment and energy consumption, Journal of Computational and Applied Mathematics 473 (2026) 116876.

[32] H. F. Rahman, M. N. Janardhanan, S. Ponnambalam, Energy aware semi-automatic assembly line balancing problem considering ergonomic risk and uncertain processing time, Expert Systems with Applications 231 (2023) 120737.

[33] Z. Huang, H. Zhang, G. Tian, M. Yang, D. Wang, Z. Li, Energy-efficient human-robot collaborative U-shaped disassembly line balancing problem considering turn on-off strategy: Uncertain modeling and solution method, Journal of Manufacturing Systems 80 (2025) 38–69.

[34] D. Battini, X. Delorme, A. Dolgui, A. Persona, F. Sgarbossa, Ergonomics in assembly line balancing based on energy expenditure: a multi-objective model, International Journal of Production Research 54 (3) (2016) 824–845.

[35] P. Jiang, Z. Wang, X. Li, X. V. Wang, B. Yang, J. Zheng, Energy consumption prediction and optimization of industrial robots based on LSTM, Journal of Manufacturing Systems 70 (2023) 137–148.

[36] Z. Bi, L. Wang, Optimization of machining processes from the perspective of energy consumption: A case study, Journal of Manufacturing Systems 31 (4) (2012) 420–428.

[37] K. Gao, Y. Huang, A. Sadollah, L. Wang, A review of energy-efficient scheduling in intelligent production systems, Complex & Intelligent Systems 6 (2) (2020) 237–249.

[38] A. Missaoui, C. Ozturk, B. O'Sullivan, M. Garraffa, Energy-efficient manufacturing scheduling: A systematic literature review, in: A. J. Kulkarni, A. H. Gandomi (Eds.), Handbook of Formal Optimization, Springer Nature Singapore, 2023, p. 1309–1334.

[39] L. Xu, P. Liu, X. Wang, Y. Lin, Y. Shi, X. Huang, X. V. Wang, Energy flexible manufacturing systems: A systematic literature review, trends and challenges, Journal of Manufacturing Systems 82 (2025) 83–97.

[40] A. Kovács, Z. Szádoczki, D. Karnok, Towards explainable optimization of production system configurations, IFAC-PapersOnLine 59 (10) (2025) 482–487, 11th IFAC Conference on Manufacturing Modelling, Management and Control (IFAC MIM 2025).

[41] D. Tsutsumi, J. Bergmann, P. Dobrovoczki, N. Hayashi, A. Kovács, Z. Szádoczki, Ádám Szaller, S. Umeda, Dynamic production line rebalancing by alternative plans for compensating equipment failures, Procedia CIRP 120 (2023) 810–815.

[42] D. Tsutsumi, J. Bergmann, P. Dobrovoczki, N. Hayashi, A. Kovács, Z. Szádoczki, Ádám Szaller, Managing fluctuations in production via an optimal portfolio of assembly line configurations, Procedia CIRP 130 (2024).

[43] P. Dobrovoczki, A. Kovács, H. Sakata, D. Tsutsumi, Integrated system configuration and layout planning for flexible manufacturing systems, Journal of Manufacturing Systems 77 (2024) 384–397.

[44] Bosch Rexroth AG, Training systems for automation (R999000216, version 1-1 EN), product documentation (2018).

[45] M. Roesch, C. Berger, S. Braunreuther, G. Reinhart, Cost-model for energy-oriented production control, in: 2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), 2018, pp. 158–162.

Appendix A. Data for the mMS Use Case

Component name	Task name	Dur (s)	Compressed air flow (Nl/s)
Separation magazine	T1: Load lower part	4.5	≈ 0
Conveyor	T2: Move lower part to Loc1.2	3.8	0
Shape control unit	T3: Check lower part	7.0	≈ 0
Conveyor	T4: Move lower part to S2	3.8	0
Separation magazine	T5: Load upper part	4.5	≈ 0
Conveyor	T6: Move upper part to Loc1.2	3.8	0
Shape control unit	T7: Check upper part	7.0	≈ 0
Conveyor	T8: Move upper part to S2	3.8	0
Conveyor	T9: Move lower part to Loc2.1	3.5	0
Pinning unit	T10: Insert pins into lower part	9.0	0.035
Conveyor	T11: Move lower part to Loc2.2	3.5	0
Press	T12: Take lower part into press	8.0	0.13
Conveyor	T13: Move upper part to Loc2.2	7.0	0
Press	T14: Join upper+lower parts	8.0	0.264
Conveyor	T15: Move product to S3	5.5	0
Conveyor	T16: Move product to Loc3.1	8.0	0
Warehouse	T17: Store product	15.0	0.029

Table A.4: The process plan of the single product and the compressed air consumption of the appropriate component-task pairs in the sample use case.

Modified task	Modification	Medium	High
T3: Check lower part	Additional reject cylinder movements	9	13
T7: Check upper part	Additional reject cylinder movements	9	13
T10: Insert pins into lower part	Additional product holder movement	5	7
T12: Take lower part into press	Vacuum lifter is on for additional time (sec)	1	2
T14: Join upper+ lower parts	Vacuum lifter is on for additional time (sec)	1	2
T17: Store product	Acceleration of the warehouse is increased (mm/sec)	300	800
T17: Store product	Velocity of the warehouse is increased (mm/sec)	1500	3000

Table A.5: Modifications of process parameters for the medium- and high-complexity variants compared to the original low-complexity variant used for the verification of the energy consumption model.

Appendix B. Detailed Computational Results

nStation	nTask	nComponent	Opt	Avg. time (s)	Avg. gap	Max. gap
6	68	18	5	0.1	0.0%	0.0%
		36	5	0.2	0.0%	0.0%
		54	5	0.4	0.0%	0.0%
	136	18	5	0.1	0.0%	0.0%
		36	5	0.4	0.0%	0.0%
		54	5	0.9	0.0%	0.0%
	204	18	5	0.1	0.0%	0.0%
		36	5	0.5	0.0%	0.0%
		54	5	4.7	0.0%	0.0%
12	136	18	5	0.1	0.0%	0.0%
		36	5	0.1	0.0%	0.0%
		54	5	0.2	0.0%	0.0%
	272	18	5	0.1	0.0%	0.0%
		36	5	0.2	0.0%	0.0%
		54	5	0.6	0.0%	0.0%
	408	18	5	0.1	0.0%	0.0%
		36	5	0.7	0.0%	0.0%
		54	5	1.0	0.0%	0.0%
18	204	18	5	0.2	0.0%	0.0%
		36	5	0.3	0.0%	0.0%
		54	5	0.3	0.0%	0.0%
	408	18	5	0.3	0.0%	0.0%
		36	5	0.3	0.0%	0.0%
		54	5	0.9	0.0%	0.0%
	612	18	5	0.2	0.0%	0.0%
		36	5	2.0	0.0%	0.0%
		54	5	2.8	0.0%	0.0%
			total: 135	avg.: 0.7	avg.: 0.0%	max.: 0.0%

Table B.6: Computational results on generated problem instances: investment cost minimized ($w_I = 1$, $w_E = 0$), compressed air supplied externally.

nStation	nTask	nComponent	Opt	Avg. time (s)	Avg. gap	Max. gap
6	68	18	5	0.1	0.0%	0.0%
		36	5	0.3	0.0%	0.0%
		54	5	2.2	0.0%	0.0%
	136	18	5	0.2	0.0%	0.0%
		36	5	0.7	0.0%	0.0%
		54	5	2.0	0.0%	0.0%
	204	18	5	0.1	0.0%	0.0%
		36	5	1.4	0.0%	0.0%
		54	4	144.3	3.6%	18.0%
12	136	18	5	0.2	0.0%	0.0%
		36	5	0.2	0.0%	0.0%
		54	5	0.4	0.0%	0.0%
	272	18	5	0.2	0.0%	0.0%
		36	5	0.5	0.0%	0.0%
		54	5	1.2	0.0%	0.0%
	408	18	5	0.9	0.0%	0.0%
		36	1	480.8	6.0%	10.8%
		54	0	600.0	9.1%	10.6%
18	204	18	5	0.5	0.0%	0.0%
		36	5	0.7	0.0%	0.0%
		54	5	0.6	0.0%	0.0%
	408	18	5	2.9	0.0%	0.0%
		36	5	1.9	0.0%	0.0%
		54	5	1.8	0.0%	0.0%
	612	18	1	481.4	18.1%	24.8%
		36	0	600.0	7.0%	10.6%
		54	0	600.0	6.8%	8.4%
			total: 111	avg.: 108.3	avg.: 1.9%	max.: 24.8%

Table B.7: Computational results on generated problem instances: total cost minimized ($w_I = 1$, $w_E = 1$), compressed air supplied externally.

nStation	nTask	nComponent	Opt	Avg. time (s)	Avg. gap	Max. gap
6	68	18	5	0.0	0.0%	0.0%
		36	5	0.2	0.0%	0.0%
		54	5	0.7	0.0%	0.0%
	136	18	5	0.1	0.0%	0.0%
		36	5	0.7	0.0%	0.0%
		54	5	1.9	0.0%	0.0%
	204	18	5	0.0	0.0%	0.0%
		36	4	190.1	0.4%	1.8%
		54	0	600.0	5.7%	9.2%
12	136	18	5	0.1	0.0%	0.0%
		36	5	0.6	0.0%	0.0%
		54	5	8.8	0.0%	0.0%
	272	18	5	0.1	0.0%	0.0%
		36	5	1.3	0.0%	0.0%
		54	4	226.8	0.6%	3.2%
	408	18	5	0.3	0.0%	0.0%
		36	0	600.0	1.9%	2.5%
		54	0	600.0	7.7%	10.7%
18	204	18	5	0.2	0.0%	0.0%
		36	5	1.2	0.0%	0.0%
		54	5	3.6	0.0%	0.0%
	408	18	5	0.2	0.0%	0.0%
		36	5	4.7	0.0%	0.0%
		54	4	127.7	0.4%	2.1%
	612	18	5	0.3	0.0%	0.0%
		36	0	600.0	5.8%	11.5%
		54	0	600.0	8.4%	12.9%
			total: 107	avg.: 132.2	avg.: 1.1%	max.: 12.9%

Table B.8: Computational results on generated problem instances: energy cost minimized ($w_I = 0$, $w_E = 1$), compressed air supplied externally.

nStation	nTask	nComponent	Opt	Avg. time (s)	Avg. gap	Max. gap
6	68	18	5	0.2	0.0%	0.0%
		36	5	0.3	0.0%	0.0%
		54	5	0.3	0.0%	0.0%
	136	18	5	0.4	0.0%	0.0%
		36	5	3.5	0.0%	0.0%
		54	5	2.8	0.0%	0.0%
	204	18	5	0.3	0.0%	0.0%
		36	5	5.4	0.0%	0.0%
		54	5	4.9	0.0%	0.0%
12	136	18	5	9.1	0.0%	0.0%
		36	5	2.5	0.0%	0.0%
		54	5	2.8	0.0%	0.0%
	272	18	5	6.7	0.0%	0.0%
		36	5	3.4	0.0%	0.0%
		54	5	5.6	0.0%	0.0%
	408	18	5	43.5	0.0%	0.0%
		36	5	6.7	0.0%	0.0%
		54	5	7.9	0.0%	0.0%
18	204	18	5	3.1	0.0%	0.0%
		36	5	5.5	0.0%	0.0%
		54	5	8.6	0.0%	0.0%
	408	18	2	362.9	11.5%	21.5%
		36	5	6.0	0.0%	0.0%
		54	5	9.3	0.0%	0.0%
	612	18	1	480.6	6.7%	13.1%
		36	5	7.3	0.0%	0.0%
		54	5	9.8	0.0%	0.0%
			total: 128	avg.: 37.0	avg.: 0.7%	max.: 21.5%

Table B.9: Computational results on generated problem instances: investment cost minimized ($w_I = 1$, $w_E = 0$), compressed air generated within the system.

nStation	nTask	nComponent	Opt	Avg. time (s)	Avg. gap	Max. gap
6	68	18	5	0.2	0.0%	0.0%
		36	5	1.0	0.0%	0.0%
		54	5	4.7	0.0%	0.0%
	136	18	5	0.5	0.0%	0.0%
		36	5	2.3	0.0%	0.0%
		54	5	18.8	0.0%	0.0%
	204	18	5	0.5	0.0%	0.0%
		36	5	5.0	0.0%	0.0%
		54	3	245.5	3.6%	11.2%
12	136	18	5	4.1	0.0%	0.0%
		36	5	40.3	0.0%	0.0%
		54	3	253.6	1.9%	5.4%
	272	18	4	136.0	3.7%	18.4%
		36	3	242.8	1.4%	6.1%
		54	3	306.2	2.2%	5.6%
	408	18	2	435.5	10.8%	22.5%
		36	1	481.1	3.1%	4.4%
		54	1	481.8	6.1%	8.0%
18	204	18	4	147.2	1.4%	7.0%
		36	5	34.7	0.0%	0.0%
		54	3	393.5	0.4%	1.1%
	408	18	2	361.6	15.8%	39.5%
		36	0	600.0	2.6%	3.0%
		54	0	600.0	4.3%	6.0%
	612	18	0	600.0	28.0%	31.2%
		36	0	600.0	4.4%	6.4%
		54	0	600.0	5.6%	7.1%
			total: 84	avg.: 244.3	avg.: 3.5%	max.: 39.5%

Table B.10: Computational results on generated problem instances: total cost minimized ($w_I = 1$, $w_E = 1$), compressed air generated within the system.

nStation	nTask	nComponent	Opt	Avg. time (s)	Avg. gap	Max. gap
6	68	18	5	0.4	0.0%	0.0%
		36	4	297.0	2.9%	14.6%
		54	1	537.8	20.3%	44.0%
	136	18	5	0.4	0.0%	0.0%
		36	0	600.0	12.5%	23.9%
		54	0	600.0	18.4%	36.6%
	204	18	5	15.4	0.0%	0.0%
		36	0	600.0	23.9%	31.0%
		54	0	600.0	25.3%	32.9%
12	136	18	2	449.4	4.6%	11.2%
		36	0	600.0	30.8%	35.1%
		54	0	600.0	36.7%	40.8%
	272	18	0	600.0	11.6%	16.8%
		36	0	600.0	35.5%	41.5%
		54	0	600.0	39.7%	47.3%
	408	18	0	600.0	23.0%	29.6%
		36	0	600.0	37.2%	38.8%
		54	0	600.0	35.4%	38.7%
18	204	18	0	600.0	20.7%	25.6%
		36	0	600.0	34.8%	42.0%
		54	0	600.0	40.6%	46.1%
	408	18	2	360.7	15.2%	31.1%
		36	0	600.0	34.0%	44.2%
		54	0	600.0	40.7%	46.6%
	612	18	1	480.7	14.2%	25.2%
		36	0	600.0	35.6%	41.9%
		54	0	600.0	35.2%	41.4%
			total: 25	avg.: 501.5	avg.: 23.3%	max.: 47.3%

Table B.11: Computational results on generated problem instances: energy cost minimized ($w_I = 0$, $w_E = 1$), compressed air generated within the system.