Journal of Manufacturing Systems 89 (2026) 483–503

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

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

Technical paper

Simulation-based multi-objective decision support for Comprehensive Digital Twins of smart manufacturing systems

Atieh Khodadadi [a,*], Sanja Lazarova-Molnar [a,b], Xueping Li [c]

[a] Institute AIFB, Karlsruhe Institute of Technology, Karlsruhe, Germany
[b] The Maersk Mc-Kinney Moller Institute, University of Southern Denmark, Odense, Denmark
[c] Department of Industrial and Systems Engineering, University of Tennessee, Knoxville, TN, USA

ARTICLE INFO

Keywords:
Comprehensive Digital Twins
Smart manufacturing systems
Multidimensional stochastic petri nets
Multi-flow process mining
Multi-objective decision support
Simulation-based multi-objective optimization

ABSTRACT

Decision support in Smart Manufacturing Systems (SMSs) often requires balancing conflicting objectives related to multiple system dimensions, such as maximizing the number of completed products while minimizing energy consumption and CO_2 emissions. Comprehensive Digital Twins (CDTs) integrate multiple system dimensions within one executable model to support multidimensional analysis and multi-objective decision-making. However, an automatically extracted, simulation-ready CDT is not inherently optimization-ready because controllable parameters, admissible ranges, optimization objectives, and decision-to-model mappings remain unspecified. This paper addresses this gap by introducing a semi-automated framework that prepares an automatically extracted CDT for simulation-based multi-objective optimization. Starting from multidimensional event logs, Multi-Flow Process Mining extracts dimension-specific Stochastic Petri Nets and integrates them into an executable Multidimensional Stochastic Petri Net represented in Multidimensional Petri Net Markup Language. An optimization-preparation layer combines model parameter information with expert-defined selections, bounds, and objectives, and maps decision vectors to executable model configurations evaluated through simulation. The framework is demonstrated using an illustrative manufacturing line for robotic assembly and welding with rework. The Nondominated Sorting Genetic Algorithm II (NSGA-II) is applied as the primary Pareto-based optimizer, while equal-weight weighted-sum scalarization is evaluated as a comparative baseline. The NSGA-II Pareto-front approximation supports post-search scenario selection. In a 100-replication reference-model evaluation, the selected scenario reduced CO_2 emissions by 3.81%, while completed products and grid-energy consumption changed only marginally (-0.29% and $+0.17\%$, respectively). These results support the proof-of-concept feasibility of extending an automatically extracted multidimensional simulation model to multi-objective decision support.

1. Introduction

The increasing integration of sensing, connectivity, and computation into manufacturing operations is transforming production systems into cyber-physical environments [1]. Within this context, Smart Manufacturing Systems (SMSs) integrate Cyber-Physical Systems (CPS) and Information and Communication Technologies (ICT), particularly Artificial Intelligence (AI) and the Internet of Things (IoT) [2]. These technologies enable SMSs to generate large volumes of operational data that can be used for monitoring, control, performance analysis, and decision support [3]. Digital Twins (DTs) have consequently become an important means of connecting physical manufacturing systems with their digital representations to support the monitoring, analysis, and optimization of system behavior [4].

Nomenclature

Abbreviation	Definition	Abbreviation	Definition
SMS	Smart Manufacturing System	MDSPN	Multidimensional Stochastic Petri Net
CPS	Cyber-Physical System	PNML	Petri Net Markup Language
ICT	Information and Communication Technologies	MDPNML	Multidimensional Petri Net Markup Language

(*continued on next page*)

* Corresponding author.
E-mail address: atieh.khodadadi@kit.edu (A. Khodadadi).

https://doi.org/10.1016/j.jmsy.2026.09.021
Received 2 August 2026; Received in revised form 28 August 2026; Accepted 25 September 2026
Available online 1 October 2026
0278-6125/© 2026 The Author(s). Published by Elsevier Ltd on behalf of The Society of Manufacturing Engineers. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).
Open access funding enabled and organized by Projekt DEAL.

(*continued*)

Nomenclature

Abbreviation	Definition	Abbreviation	Definition
AI	Artificial Intelligence	MOSO	Multi-Objective Simulation Optimization
IoT	Internet of Things	MOEA	Multi-Objective Evolutionary Algorithm
DT	Digital Twin	NSGA-II	Nondominated Sorting Genetic Algorithm II
DES	Discrete-Event Simulation	SPEA2	Strength Pareto Evolutionary Algorithm 2
SD	System Dynamics	SMS-EMOA	S-Metric Selection Evolutionary Multiobjective Optimization Algorithm
ABS	Agent-Based Simulation	MOEA/D	Multiobjective Evolutionary Algorithm Based on Decomposition
HS	Hybrid Simulation	NSGA-III	Nondominated Sorting Genetic Algorithm III
PM	Process Mining	GA	Genetic Algorithm
SPN	Stochastic Petri Net	KPI	Key Performance Indicator
CPN	Colored Petri Net	CI	Confidence Interval
MFPM	Multi-Flow Process Mining	MAG	Metal Active Gas
CDT	Comprehensive Digital Twin		

As Fig. 1 illustrates, a DT comprises three primary components: (i) the physical entities, (ii) their virtual counterpart, and (iii) bidirectional data exchange between them [5]. This bidirectional data exchange supports ongoing data acquisition, DT model validation, and knowledge extraction, keeping the DT synchronized with its physical counterpart [6]. DTs are typically realized through computational models, such as analytical equations, machine-learning models, simulation models, or hybrid models, depending on the intended purpose of the twin [7,8]. In manufacturing, simulation-based models are particularly valuable because they provide an explicit and executable representation of system behavior, in contrast to purely black-box input-output models [9]. This supports structural understanding, what-if analysis, and model-based decision support, especially for production systems characterized by arrivals, routing decisions, machine operations, failures, repairs, and resource interactions. DTs and data-driven simulation models have therefore become increasingly important in industrial applications, where they are used to improve productivity, reduce costs, and enhance resource efficiency [4]. In this paper, we therefore adopt a simulation model as the virtual counterpart of the DT.

There are several modeling formalisms for representing the virtual counterpart of SMSs. Among the available Petri-net-based formalisms, Stochastic Petri Nets (SPNs) [11] provide a suitable foundation for representing event-driven manufacturing systems. SPNs offer a well-established graphical and mathematical formalism for representing concurrency, synchronization, resource sharing, and stochastic system behavior. In addition, SPN models are well-suited to data-driven DT development, since Process Mining (PM) [12] techniques combined with data analysis can be used to discover their control-flow structure and estimate their stochastic execution semantics from event logs generated by the underlying manufacturing system [13]. The resulting models can then be executed by simulation and used for temporal performance evaluation, what-if analysis, and model-based decision support. As such, SPNs provide a rigorous and executable basis for representing the dynamic behavior of processes in SMSs.

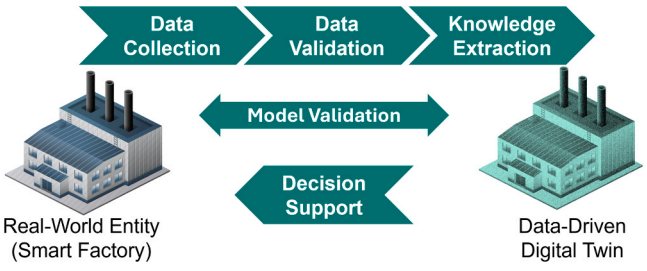


Fig. 1. Constituent parts of a Digital Twin setup for smart manufacturing systems [10].

Manufacturing decision-making is increasingly framed as a multi-objective problem that must balance productivity-oriented goals with other objectives such as energy efficiency, carbon footprint reduction, and waste minimization. This shift toward multidimensional manufacturing decision-making, motivated by Industry 4.0 [1], is further strengthened by the Industry 5.0 perspective [14], which emphasizes sustainability and resilience alongside competitiveness. This shift is also motivated by policy and market pressures, including the European Climate Law [15], which establishes a binding target of at least a 55% net reduction in greenhouse-gas emissions by 2030 relative to 1990 and climate neutrality by 2050 [16], and the U.S. Department of Energy (DOE) Industrial Decarbonization Roadmap [17]. These pressures require manufacturing decisions to account for production and environmental performance simultaneously, motivating data-driven Digital Twin models that represent multiple dimensions, including time, energy, emissions, and waste. In many existing approaches, additional non-temporal performance quantities are either represented through reward- or measure-based constructions [18,19] rather than as explicit multidimensional effects in the executable model or incorporated manually during model development.

In our previous work, we addressed this limitation by proposing Multi-Flow Process Mining (MFPM), an extension of PM that enables the automatic extraction of dimension-specific (unidimensional) SPN models from multidimensional event logs [20]. In a subsequent study [21], we introduced the Comprehensive Digital Twin (CDT) framework for SMSs, which integrates unidimensional SPNs into a unified Multidimensional Stochastic Petri Net (MDSPN). The resulting MDSPN preserves an explicit control-flow structure while associating each activity with its multidimensional impacts, thereby supporting cross-dimensional system analysis. In our preceding *Journal of Manufacturing Systems* study [22], we further developed and validated the automated extraction of MDSPN-based CDTs from multidimensional event logs. The automated extraction procedure generates simulation-ready executable MDSPN models for multidimensional analysis. We subsequently introduced Multidimensional Petri Net Markup Language (MDPNML) [23] to represent the extracted MDSPN structure and its multidimensional simulation parameters in an executable interchange format. These previous contributions provide the simulation-ready CDT foundation adopted in the present study.

However, an automatically extracted and simulation-ready MDPNML model (or MDSPN model) is not directly optimization-ready. Although the extracted model can reproduce and analyze the multidimensional behavior represented in an SMS, it does not by itself determine which model parameters should be treated as controllable decisions, how a numerical decision vector should be mapped to the corresponding MDPNML elements, or which modeled dimensions should be used as optimization objectives. Some optimization-related information is explicitly available in the extracted model. For example, the selection or routing decisions can be exposed for optimization through their MDPNML representation. Other decisions, such as available device speed settings and technical operating limits, cannot be inferred reliably from the generated data or model structure alone and must be provided by domain experts or additional technical documentation. A methodological gap, therefore, remains between the automatic extraction of a simulation-ready CDT and its systematic preparation for simulation-based multi-objective optimization.

To address this gap, this paper introduces the concept of an optimization-ready Comprehensive Digital Twin. The proposed approach extends the automatically extracted MDSPN–MDPNML pipeline with a semi-automated optimization-preparation layer. The layer identifies optimization-relevant parameters represented in the model, incorporates expert-defined parameter selections and admissible ranges, and establishes an executable mapping between a decision vector and the corresponding MDPNML elements. During optimization, each candidate solution is automatically translated into a corresponding MDPNML model configuration and evaluated through simulation. The resulting multidimensional performance values are then returned to the optimization algorithm. The semi-automated characterization reflects the division between information available from the extracted model and information requiring expert specification: model-represented parameters, such as selection weights, can be processed systematically, whereas decisions requiring physical, technical, or managerial information remain expert-defined. The contribution is therefore not the fully automatic generation of an optimization problem, but the systematic combination of automatically extracted multidimensional behavior with model-derived and externally provided decision information.

The proposed methodology focuses on preparing an automatically extracted CDT for simulation-based multi-objective optimization. Among the available approaches, this study adopts Multi-Objective Evolutionary Algorithms (MOEAs) [24]. The purpose is to provide stakeholders with a set of feasible Pareto-efficient operating configurations rather than a single solution based on predefined objective priorities. A configuration is Pareto-efficient when none of its objectives can be improved without worsening at least one other objective; the objective values of all such configurations form the Pareto front [25]. The resulting Pareto-front approximation makes the trade-offs among objective values derived from the modeled dimensions and KPIs explicit, including completed products, energy consumption, CO_2 emissions, and waste. Stakeholders can then select a suitable configuration according to their current priorities and operational requirements. Weighted-sum scalarization [26] is additionally implemented as a comparative baseline, while surrogate-assisted optimization [27] remains a relevant alternative for computationally expensive applications.

Building on the simulation-ready CDT foundation, the present study extends the workflow to the systematic preparation and integration of the extracted CDT for simulation-based multi-objective optimization. In contrast to DT-based approaches developed for specific manufacturing optimization applications [28–30], our framework combines automated data-driven model extraction with the semi-automated preparation of the extracted model for simulation-based multi-objective optimization. The contributions of this paper are threefold. First, we introduce a methodology for transforming an automatically extracted MDSPN-based CDT from simulation-ready to optimization-ready by combining model-derived information with expert-defined information on controllability, admissible ranges, objectives, and constraints. Second, we establish a traceable decision-to-model mapping that translates decision vectors into executable MDPNML configurations and returns the resulting objective values and, where defined, constraint values to the optimizer. Third, we demonstrate the complete workflow in an illustrative manufacturing-line case study. The demonstration includes automated model extraction, model validation, optimization preparation, and search using the Nondominated Sorting Genetic Algorithm II (NSGA-II) [31], reevaluation of the search-stage nondominated configurations with 100 independent replications, post-search scenario selection, and evaluation of the selected scenario in the reference model. A weighted-sum scalarization baseline is additionally evaluated under equivalent case-study settings. The case study comprises two robots, two primary Metal Active Gas (MAG) welding machines, and one rework welding machine. Seven controllable parameters are optimized to obtain a Pareto-front approximation for grid-energy consumption, CO_2 emissions, and the number of completed products, after which stakeholder preferences are applied to select an operating scenario.

The remainder of this paper is organized as follows. Section 2 reviews the relevant background on Digital Twins, simulation, PM, Stochastic Petri Nets, and multi-objective optimization. Section 3 presents the proposed CDT-based decision-support methodology. Section 4 introduces the illustrative case study and the corresponding experimental results. Section 5 discusses the implications and limitations of the findings. Section 6 summarizes the study and outlines directions for future work.

2. Related Work and Background

This section reviews the theoretical and methodological background underlying the proposed CDT-based decision-support framework, covering modeling and simulation for SMSs, DTs, PM, SPNs, MOEAs, NSGA-II, and simulation-based multi-objective optimization.

2.1. Digital Twins and Discrete-Event Simulation for Manufacturing Systems

Simulation replicates the behavior of a real-world process or system over time by creating and analyzing a synthetic history through a representative model [32]. In the context of DTs, simulation serves as the computational foundation for prediction, optimization, and decision-making support. Simulation modeling encompasses a broad range of formalisms and approaches. In manufacturing and DT research, commonly used paradigms include Discrete-Event Simulation (DES) [33], System Dynamics (SD) [34], Agent-Based Simulation (ABS) [35], and Hybrid Simulation (HS) [36]. DES represents systems whose states change at discrete event times. Manufacturing systems exhibit event-driven behavior in which events, such as part arrivals, operation completions, breakdowns, and repairs, occur at discrete points in time and may change the system state. Consequently, DES is one of the most widely used methods for studying manufacturing processes [37].

Michael Grieves initially presented the concept currently referred to as the DT in 2002 during an industry discourse on product lifecycle management [38]. DTs are utilized in several sectors, such as aerospace, energy, healthcare, and particularly in SMSs, where they facilitate real-time monitoring, predictive analytics, and data-driven optimization [39]. Several DT frameworks for SMSs have been proposed in the literature. A prominent example is the framework proposed by Friederich et al. [40], which supports the automated generation of simulation models as the basis for DTs in smart factories. In this paper, we adopt that general view as the foundation for developing CDTs of SMSs, while extending it toward multidimensional modeling and multi-objective decision support.

As illustrated in Fig. 2, data-driven DT development starts with the identification of the relevant entities (including machines, production lines, and control elements) and the establishment of pipelines that continuously capture data (such as diagnostics and logs/traces) from sensors, operator inputs, and the factory's internal network. The acquired data are integrated and stored in a unified format. Model accuracy depends on data quality; the subsequent data-validation phase [41] therefore includes validity checks, noise removal, treatment of missing values, and anomaly detection (either supervised or unsupervised) to identify deviations from typical behavior. The workflow subsequently performs knowledge extraction: (i) event detection and labeling [42], employing machine learning techniques such as classification or clustering to identify simulation-relevant events, with expert verification of labels, and (ii) process discovery from event logs, which is employed to generate an initial, integrated simulation model connected to live data streams for automatic updates as the system changes. The model is subjected to ongoing validation by assessing its performance against the

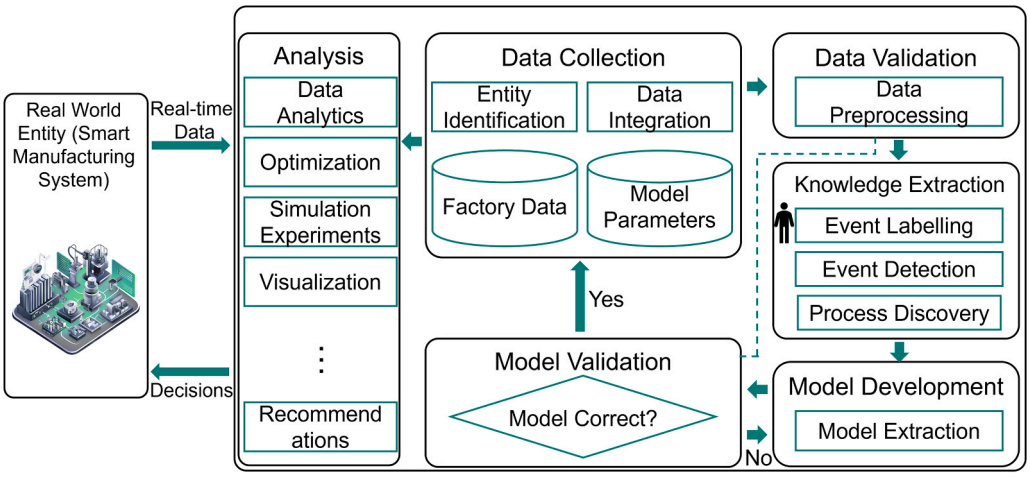


Fig. 2. Framework for data-driven Digital Twins of smart manufacturing systems [40].

system's established Key Performance Indicators (KPIs), including throughput and the number of completed products. After validation, the DT can be used to perform simulation experiments and explore what-if scenarios. Finally, simulation results are assessed against established KPIs to provide stakeholders with information for manufacturing-system improvement.

2.2. Process Mining (PM)

PM techniques [43] analyze event logs to discover process models, check conformance between event data and expected behavior, and enhance models with performance information. The capabilities of PM have inspired applications in manufacturing, healthcare, logistics, and business process management, where understanding actual operational behavior becomes essential. PM techniques are typically classified into three categories: process discovery (extracting a model from the event log), conformance checking (evaluating logs against an existing model to identify discrepancies), and enhancement (improving the model based on PM findings).

Event logs constitute the basis of PM and record temporally ordered events, each containing a case identifier, activity label, and timestamp. We adopt the following formalization [44]: Let E denote the set of events. A trace is a finite ordered sequence of events, $\sigma = \langle e_1, \ldots, e_n \rangle$, where $e_i \in E$. An event log L is a finite multiset of traces. Each event e has at least three standard attributes: activity label (Event Label) $\alpha(e)$, case identifier (ID) $\gamma(e)$, and timestamp (Time Stamp) $\tau(e)$. Events with the same case identifier are grouped into traces by ordering the corresponding events according to their timestamps.

PM reveals how processes are actually executed and how their observed behavior differs from the expected process [45]. Depending on its purpose and data connection, a manufacturing DT may be synchronized continuously, periodically, or on demand with the physical system. In each case, the reliability of its analyses depends on maintaining a model that is sufficiently current and valid for the intended decision task. However, conventional modeling methodologies frequently depend on idealized process descriptions or design documents that become obsolete when the system undergoes modifications. PM addresses this limitation by discovering process models from event data generated by the actual system [46], which can then be utilized for DT model extraction. For instance, Pourbafrani and van der Aalst [47] define a reference meta model that maps event log information to DES parameters. Castiglione [48] uses a PM approach to enable automated generation of digital models for manufacturing systems. Kumbhar et al. [49] similarly combine PM with a simulation-based Digital Twin for bottleneck detection, diagnosis, and throughput improvement in manufacturing systems.

2.3. Stochastic Petri Nets (SPNs)

Discrete event systems, including manufacturing systems, can be represented using various modeling formalisms. Among these, Petri nets provide a unified graphical and mathematical framework for modeling, analysis, and design, and are well suited to systems exhibiting concurrency, synchronization, asynchrony, resource sharing, and conflict [50]. For instance, Mejía and Pereira [51] use Petri nets to formulate and solve multi-objective scheduling problems in flexible manufacturing systems.

Petri nets have many well-studied extensions that primarily differ in expressive power and in the additional semantics and annotations attached to net elements, such as data, time, and stochastic behavior [52], including Colored Petri Nets (CPNs) [53] and SPNs [54]. In this study, we adopt SPNs because SPN models can be constructed from event logs by combining PM for control flow discovery with data analysis for estimating stochastic firing behavior [55]. A step-by-step description of the data-driven automatic SPN extraction workflow is provided in our previous work [13]. We adopt the formal definition of SPNs provided by Lazarova-Molnar [56]. An SPN is defined as a tuple $SPN = (P, T, A, G, m_0)$, where:

- P is a finite set of places (drawn as circles), $P = \{P_1, P_2, \ldots, P_m\}$,
- T is a finite set of transitions (drawn as bars) with associated firing distributions or weights, $T = \{T_1, T_2, \ldots, T_n\}$,
- $A = A^I \cup A^O \cup A^H$is the set of input, output, and inhibitor arcs with multiplicities; $A^I = \left\{a_1^i, a_2^i, \ldots, a_j^i\right\}$ connect places to transitions; $A^0 = \left\{a_1^o, a_2^o, \ldots, a_k^o\right\}$ connect transitions to places, and $A^H = \left\{a_1^h, a_2^h, \ldots, a_i^h\right\}$ prevent a transition from firing if a certain number of tokens are present in the corresponding inhibitor place. $A^O \subseteq (T \times P \times N)$ and $A^H, A^I \subseteq (P \times T \times N)$,
- $G = \{g_1, g_2, \ldots, g_r\}$ is a set of guard functions for transitions,
- and m_0 is the initial marking (token distribution in the SPN).

Arc multiplicities specify how many tokens are consumed from input places and produced in output places, while guard predicates impose additional logical enabling constraints. A transition is enabled when the following conditions hold: (i) each input place contains at least the number of tokens required by the corresponding input arc multiplicities, (ii) no inhibitor arc disables the transition, that is, each inhibitor place satisfies the blocking condition induced by its inhibitor multiplicity threshold, and (iii) all guard predicates associated with the transition evaluate to true. Once enabled, firing occurs according to the net's firing rules and updates the marking from M to M' by consuming and producing tokens as prescribed by the arc multiplicities.

2.4. Simulation-based Multi-objective Optimization

In many engineering applications, system performance cannot be expressed in closed form as an explicit function of the decision variables. Instead, performance is assessed by running a simulation model under a given decision vector and computing one or more performance measures from the simulation output [57]. This setting is commonly referred to as "simulation optimization" or "optimization via simulation", where objective values and, where applicable, constraint values are estimated through simulation rather than obtained from closed-form analytical expressions [57].

When multiple conflicting performance measures are considered simultaneously (such as increasing the number of completed products while decreasing energy consumption), the problem is referred to as Multi-Objective Simulation Optimization (MOSO) [58]. In stochastic simulation, the outcome of objective $i$for a decision vector $x$can be represented by the random variable $Y_i(x,\xi)$, where ξdenotes the simulation randomness. The corresponding objective function is commonly defined as the expected performance $f_i(x) = \mathbb{E}[Y_i(x,\xi)]$. Because this expectation is generally unknown, it is estimated using R simulation replications:

$$\widehat{f}_i(x) = \frac{1}{R}\sum_{r=1}^{R} Y_i(x,\xi_r).$$

Consequently, the optimizer operates on objective estimates affected by sampling error. The number and allocation of replications, as well as the reevaluation of competitive solutions, are therefore important when constructing and interpreting a Pareto-front approximation [58]. Formally, a multi-objective optimization problem seeks $x \in \Omega \subseteq \mathbb{R}^n$that optimizes $F(x) = (f_1(x), \ldots, f_m(x))$, where $\Omega = \{x \in X | g(x) \leq 0,\ h(x) = 0\}$ denotes the admissible decision space [59]. Here, $g(x) \leq 0$ represents additional constraints (such as requiring a KPI to remain above or below a specified threshold). In simulation-based optimization, each $f_i(x)$ is typically an expected performance measure estimated from one or more simulation replications at x [57]. Assuming minimization of all objectives, x Pareto dominates y (written $x \prec y$) if $f_i(x) \leq f_i(y)$ for all $i \in \{1, \ldots, m\}$ and $f_j(x) < f_j(y)$ for at least one j. A feasible solution is Pareto optimal if it is not dominated by any other feasible solution. In a finite set $S \subseteq \Omega$ (e.g., a population), a solution is called nondominated if no other element of S dominates it [59]. Accordingly, optimization algorithms generally produce a finite Pareto-front approximation rather than the exact Pareto front.

Simulation-based multi-objective optimization approaches differ in how stakeholder preferences are incorporated and how candidate solutions are evaluated. Scalarization methods, such as the weighted-sum and ε-constraint methods, transform the multi-objective problem into one or more parameterized single-objective problems [59,60]. Scalarization methods are useful when the relative importance of the objectives or acceptable limits for selected objectives can be specified before optimization. However, each parameter setting produces a preference-dependent solution, and several settings are required to represent different regions of the trade-off space. In addition, linear weighted-sum scalarization cannot recover unsupported Pareto-optimal solutions in non-convex regions of the Pareto front [61–63].

Pareto-based approaches instead seek multiple nondominated solutions before final stakeholder preferences are introduced. Multi-Objective Evolutionary Algorithms (MOEAs) are frequently used for this purpose because their population-based search enables several trade-off solutions to be explored simultaneously [31]. The resulting approximation provides an overview of the relationships among the objectives, after which stakeholder preferences can be applied to select an operating configuration. When individual simulation evaluations are computationally expensive, either scalarization-based or Pareto-based optimization can be supported by surrogate models [64]. A surrogate (e.g., a Gaussian-process/Kriging model) is fitted to simulation data and used to guide the selection of new design points, thereby reducing the number of expensive simulation evaluations [65]. Surrogate/model-based MOSO methods are particularly effective when evaluations are computationally expensive, but their performance depends on surrogate fidelity and the chosen acquisition criterion, and can degrade under noisy simulation outputs [66]. Promising solutions must also be verified using the original simulation model because inaccurate surrogate predictions can misdirect the optimization search [65,67].

In the present study, a direct Pareto-based evolutionary approach is adopted as the primary optimization approach because the aim is to generate a set of trade-off configurations before stakeholder preferences are specified. Candidate configurations are evaluated directly using the executable MDSPN, whose computational cost remains manageable for the considered case study. Weighted-sum scalarization is additionally implemented as a comparative baseline under equivalent case-study settings, while surrogate-assisted optimization remains a relevant alternative for computationally expensive applications.

2.5. Multi-Objective Evolutionary Algorithms (MOEAs)

In manufacturing systems, decision-making often requires balancing multiple conflicting objectives, such as productivity, energy consumption, and environmental impact. The resulting optimization problems are frequently nonlinear, non-convex, or non-differentiable, which reduces the applicability of classical gradient-based optimization methods [68]. In such cases, metaheuristic optimization methods, including evolutionary algorithms [68], simulated annealing [69], and tabu search [70], are widely used. Among these, MOEAs are particularly suitable because they do not usually require gradient information and can approximate a set of trade-off solutions in a single run, which makes them attractive for complex simulation-based and engineering decision problems [71,72].

A MOEA is a population-based evolutionary search method for multi-objective optimization [24]. Instead of seeking a single optimum, it evolves a set of candidate solutions over successive generations in order to approximate the Pareto-optimal set in the decision space and, correspondingly, the Pareto front in the objective space. This population-based nature is one of the main reasons MOEAs are effective for multi-objective problems, since multiple trade-off solutions can be explored and improved simultaneously within one run [24,60].

Most MOEAs share the same high-level workflow: (i) initialize a population of decision vectors, (ii) generate offspring by mating selection and variation operators, (iii) evaluate candidate solutions, and (iv) apply a survival-selection mechanism to construct the next population. What mainly distinguishes one MOEA from another is the way it performs survival selection and diversity preservation, how it handles constraints, and the search principle used to guide selection. Examples include Pareto-dominance-based methods, such as NSGA-II [31] and SPEA2 [73], indicator-based methods, such as SMS-EMOA [74], decomposition-based methods, such as MOEA/D [75], and reference-direction-based methods, such as NSGA-III [76]. Thus, MOEAs are similar at the framework level but differ in the selection and diversity-management mechanisms used to approximate the Pareto front. The detailed MOEA methodology and its integration into the CDT framework are presented in Section 3.6.

For problems with a small number of objectives, Pareto-based MOEAs such as NSGA-II are widely used reference methods; when the number of objectives increases beyond three, many-objective approaches such as NSGA-III, as well as indicator- or decomposition-based methods, are often considered [76,77]. In the present paper, the generic MOEA framework is instantiated using NSGA-II because the case study considers three objectives: grid-energy consumption, CO_2 emissions, and the number of completed products.

2.5.1. Nondominated Sorting Genetic Algorithm II (NSGA-II)

NSGA-II is one of the most widely adopted algorithms in the litera-

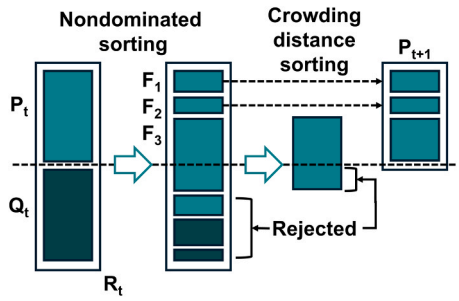


Fig. 3. NSGA-II procedure [31].

ture [78]. The widespread adoption of NSGA-II is mainly due to three features: fast nondominated sorting, elitist environmental selection, and crowding-distance-based diversity preservation. Together, these mechanisms enable NSGA-II to maintain a well-distributed approximation of the Pareto set while remaining computationally efficient [79]. NSGA-II maintains, at generation t, a parent population $P_t = \{x_1, \ldots, x_N\}$f and generates an offspring population Q_t (size N) via mating selection and variation operators (crossover and mutation). Elitism is enforced by forming the combined population $R_t = P_t \cup Q_t$ with $|R_t| = 2N$, and constructing the next parent population P_{t+1} from R_t using (i) Pareto rank via nondominated sorting and (ii) crowding distance for diversity preservation. Mating selection is typically performed by binary tournament using the crowded-comparison operator, which prefers lower rank and breaks ties in favor of a larger crowding distance [31].

Fig. 3 summarizes NSGA-II's elitist environmental selection in one generation. The parent and offspring populations are merged into R_t, partitioned into nondominated fronts $F_1, F_2, \ldots$by fast nondominated sorting, and P_{t+1} is filled by accepting complete fronts in increasing rank order. If the last accepted front must be truncated, the remaining slots are selected using the crowding distance (defined below) to preserve diversity.

Fast nondominated sorting. For each solution $p \in R_t$, define the domination count $n_p = |\{q \in R_t | q \prec p\}|$ and the dominated set $S_p = \{q \in R_t | p \prec q\}$. The first front is $F_1 = \{p \in R_t | n_p = 0\}$; subsequent fronts $F_2, F_3, \ldots$ are obtained iteratively: for each $p \in F_k$, decrease n_q by one for all $q \in S_p$; any solution whose domination count becomes zero is assigned to the next front. This yields the rank assignment $r(p) = k$ for $p \in F_k$.

Crowding distance and selection. Within each front F_k, a crowding distance $d(\cdot)$ is computed to promote diversity. For each objective i, solutions in F_k are sorted by f_i; boundary solutions are assigned $d = \infty$. For each interior solution x_j in the sorted order, update

$$d(x_j) \; += \; \frac{f_i(x_{j+1}) - f_i(x_{j-1})}{f_i^{\max} - f_i^{\min}},$$

where $f_i^{\max}$ and $f_i^{\min}$ are computed over the current front F_k (If $f_i^{\max} = f_i^{\min}$, the contribution of objective i is taken as zero to avoid division by zero.) The next population P_{t+1} is filled by adding entire fronts $F_1, F_2, \ldots$ in increasing rank order until the next front would exceed size N. If truncation is needed in the last accepted front, the remaining slots are filled by selecting solutions with the largest crowding distance (i.e., in decreasing $d(\cdot)$) [31].

2.6. Related Work on DT- and Petri-Net-Based Multi-Objective Optimization

Recent manufacturing studies have coupled DTs with multi-objective optimization for production-line configuration, reconfiguration, scheduling, and system improvement. For example, Ali et al. [28] apply a DT-supported NSGA-II framework to configuration selection in a reconfigurable manufacturing transfer line. Zhu et al. [29] propose a DT-driven reconfiguration method for human–robot collaborative manufacturing and solve the resulting multi-objective task-allocation problem using NSGA-II. Wu et al. [80] develop a DT-based green dynamic scheduling and visual-monitoring approach for a micro-hole drilling production line that considers production time and energy consumption. Dai et al. [81] develop a DT-based automated guided vehicle scheduling framework using an improved NSGA-II under energy constraints.

Related simulation-based manufacturing research has also addressed production and sustainability trade-offs. Schmitt et al. [82] combine DES, multi-objective optimization, and knowledge discovery to investigate energy-efficiency–productivity trade-offs and derive actionable decision rules for an industrial automated assembly line. Guo et al. [30] combine DT-based flexible cellular manufacturing, simulation, and multi-objective optimization to address coupled production-line problems involving layout, process capability, logistics, and equipment testing.

More broadly, Petri-net–based simulation models have also been coupled with MOEAs in several domains. For example, Zan et al. [83] propose a Pareto-based genetic algorithm for multi-objective scheduling of automated manufacturing systems, in which feasibility is checked and repaired through the underlying Petri-net model and deadlock controller. In addition, Mejía and Pereira [51] formulate multi-objective scheduling within a Petri-net framework and compare three solution procedures: a weighted genetic algorithm, NSGA-II, and a multi-objective local search approach. Amodeo et al. [84] evaluate supply-chain performance by simulating a (deterministic/stochastic) Petri-net model and use NSGA-II to search for inventory policies that trade off inventory cost and service level. Similarly, Sharda et al. [85] present a manufacturing system design framework that combines NSGA-II, Petri nets, and Bayesian uncertainty representation for robust manufacturing system configuration design.

These studies demonstrate the relevance of combining DTs, simulation models, Petri nets, and multi-objective optimization. Although several studies define application-specific decision variables, objectives, and constraints, these elements are formulated for models developed or configured for the respective application. The reviewed studies do not address the systematic preparation of an automatically extracted multidimensional executable model by combining model-derived and expert-provided decision information and mapping the resulting decision vector to the corresponding model elements. This study addresses this gap by transforming an automatically extracted, simulation-ready multidimensional model into an optimization-ready CDT. Its explicit decision-to-model mapping enables traceable evaluation of candidate operating configurations to inform stakeholder decisions.

3. CDT-Based Decision-Support Methodology

This section presents the proposed CDT-based decision-support methodology. Specifically, we introduce MDSPNs and MFPM, describe the CDT and its automated extraction process, define the validation protocol and evaluation criteria, and then present the integration of the optimization-ready CDT with an MOEA-based optimization layer and a weighted-sum scalarization workflow for simulation-based multi-objective decision support.

3.1. Multidimensional Stochastic Petri Nets (MDSPNs)

Classical SPNs model discrete-event control behavior with stochastic firing delays, while many operational and performance quantities are commonly specified via reward/measure definitions (such as reward rates or impulse rewards) layered on the underlying stochastic process rather than as the behavioral flows [86,87]. In Colored Petri Nets (CPNs), data-dependent behavior is encoded through color sets, arc inscriptions, expressions, and guards [88]. However, additional performance perspectives, such as energy consumption or emissions, are not

necessarily represented as explicit dimension-specific contributions within the executable model. Similarly, process discovery from event data primarily targets control flow, whereas additional perspectives generally require enriched event attributes or model extensions [89], which limits automated CDT development. To address this limitation, MDSPNs [90] provide a single executable model in which the behavior of system events can be represented explicitly across multiple dimensions and can be automatically discovered from multidimensional event logs using MFPM. In an MDSPN, a transition can contribute to multiple dimensions (such as time and energy) simultaneously, which supports CDT and multi-objective analysis.

In an MDSPN, each transition may affect none, some, or all tracked dimensions D. When a transition fires, it updates the marking M, exactly as in an SPN, and additionally updates a set of dimension-specific state variables (referred to as clocks in the Multidimensional Stochastic Petri Net formalism) $\left\{C^{(d)}\right\}_{d\in D}$ According to dimension impact specifications, for the temporal dimension, the update corresponds to the transition's stochastic firing delay, whereas for non-temporal dimensions, the update can be positive or negative when the dimension is modeled as a balance (e.g., waste reuse). To make these effects explicit, MDSPN extends the SPN transition definition by associating each transition t_i with a collection $\left\{\left(d, \text{type}_i^{(d)}, F_i^{(d)}\right)\right\}_{d\in D}$, where $\text{type}_i^{(d)} \in \{\text{contributing}, \text{noncontributing}\}$ and $F_i^{(d)}$ specifies the corresponding per-dimension impact. MDSPN further encodes these per-dimension effects using a segmented transition symbol and distinct clocks for the tracked dimensions.

Fig. 4 illustrates the MDSPN semantics using a robotic-arm example with three tracked dimensions: time, energy, and waste. The place *Robotic arm idle state* denotes the condition in which the robot is available but waiting for the next batch. While a token remains in this place, no processing takes place, but idle-energy consumption continues to accumulate. When a new batch arrives, transition $MT2$ fires immediately to assign the available robot to the batch and start the packaging stage. Thus, $MT2$ does not itself introduce a time delay; instead, it terminates the idle period, whose duration is given by the token's residence time in the idle-state place, and the corresponding idle-energy consumption is recorded in the energy dimension. $MT3$ then represents the packaging operation, which contributes to time, energy, and waste dimensions.

3.2. Multi-Flow Process Mining (MFPM)

MFPM [20] is applied to validated multidimensional event logs that extend a conventional event log (case identifier, activity label, timestamp) with dimension-specific attributes. In addition to the standard event log data, each event can carry measurements and descriptors for the targeted dimensions (such as energy value and energy source, waste amount and waste type), as represented by the excerpt in Table 1. Here, "NA" indicates that the attribute does not apply to the corresponding event.

As we illustrate in Fig. 5, the MFPM method proceeds in two phases.

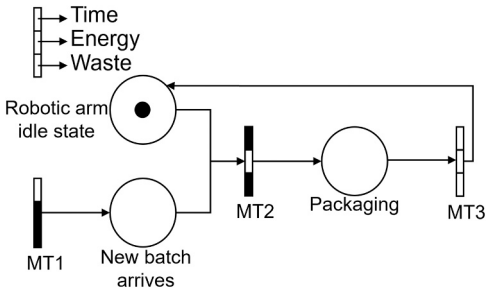


Fig. 4. Example of MDSPN modeling of a robotic arm.

Table 1
Multidimensional Event Log Excerpt.

Time Stamp	ID	Asset	Energy Stamp (kWh)	Energy Type	Waste Stamp (kg)	Waste Type	Event Label
09:22	34	NA	0.0	NA	0.0	NA	Queue End
09:22	103	Robot	40.24	Grid	0.0	NA	Robot Idle End
09:22	34	Robot	40.24	Grid	3.14	Plastic	Packing Begins
09:26	35	NA	0.0	NA	0.0	NA	New Order
…	…	…	…	…	…	…	…

First, process discovery is performed separately for each dimension to extract dimension-specific process flows and their Petri-net structure, including places, transitions, and the associated flow topology. Second, MFPM identifies the transition features for each dimension, such as firing-time distributions or weights and the corresponding dimensional updates, which are then used to parameterize the corresponding unidimensional SPN models. These unidimensional views are not combined by a simple graph union. Rather, transitions associated with the same underlying activity are aligned across dimensions to define a shared control-flow skeleton, and the corresponding dimension-specific parameters are attached to the aligned transitions. The resulting MDSPN therefore retains a single executable Petri-net structure while representing the multidimensional effects of each activity.

3.3. Automated MDSPN Extraction

In our recent work, we detailed the automated extraction of MDSPN models for representing CDTs for SMSs [22]. Building on that methodology, the extraction process begins by applying process discovery to cleaned multidimensional event logs in order to identify the system's process flow. The extracted flow is then refined by identifying relevant structural features, such as inhibitor arcs, arc multiplicities, and the multidimensional transition table, which provides a transition-level representation of the system across all modeled dimensions.

As shown in Fig. 6, the multidimensional transition table specifies, for each transition and each dimension, whether the transition is contributing or noncontributing, together with the corresponding impact type and parameter values. Depending on the semantics of the dimension, the impact type may be represented as a distribution, a rate, or a fixed value. Because the multidimensional transition table centralizes the multidimensional contribution type and impact specification of every transition, it serves as the main intermediate representation for automated model construction. On this basis, the procedure can generate either a unidimensional SPN model for a specific dimension or the full MDSPN model that integrates all considered dimensions in a single executable structure. The resulting MDSPN can then be generated in an executable form for simulation in MDPySPN [90], or exported in an interchange format. In particular, the extracted model can be represented in MDPNML [23], an extension of Petri Net Markup Language (PNML) [91] that encodes the multidimensional semantics and simulation parameters required for MDSPN-based CDTs. MDPNML preserves the PNML core structure while adding explicit dimension definitions and transition-level multidimensional impact annotations. This enables exchange and reuse of extracted multidimensional models across tools and workflows, while also supporting export or conversion to standard PNML-compatible tools when only the temporal dimension is required.

3.4. Validation Protocol and Evaluation Criteria

The extracted CDT is assessed through structural and operational validation. Structural validation examines whether the extracted model

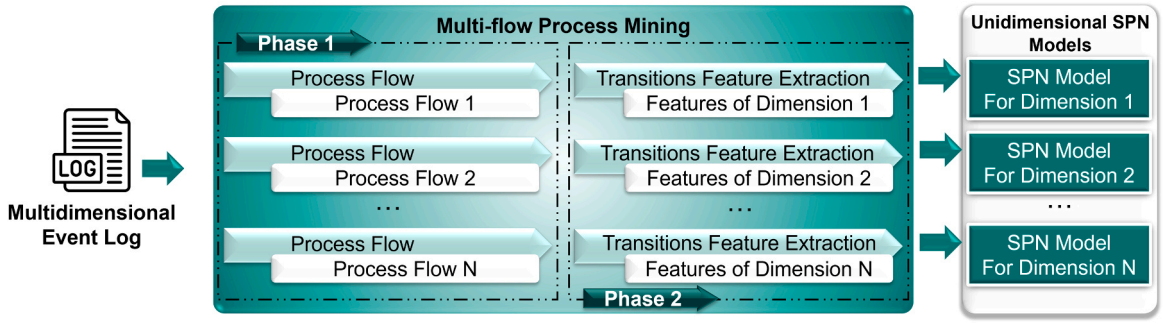


Fig. 5. Multi-flow Process Mining workflow for extracting dimension-specific SPN models from a multidimensional event log [20].

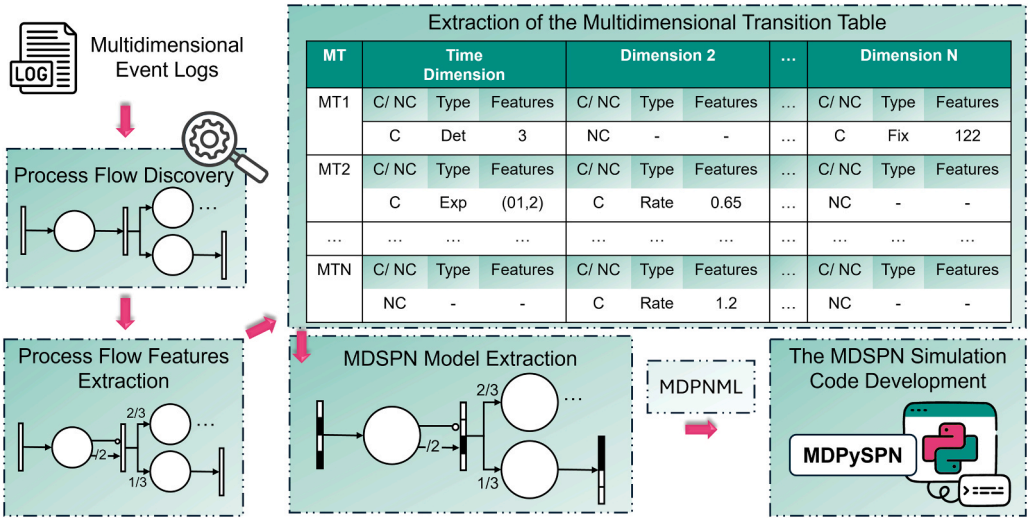


Fig. 6. Data-driven MDSPN extraction workflow [22].

structure and firing logic are consistent with the target system, whereas operational validation compares KPI values from the extracted model and the reference system under the same experimental conditions [92].

The KPI set is selected according to the modeled dimensions of the CDT. For the time dimension, representative manufacturing KPIs include the number of arriving orders, the number of completed products, and the throughput, defined here as the number of completed products divided by the number of arriving orders. For additional dimensions, the KPI set is extended with the corresponding dimensional measures, such as total and per-asset energy consumption for the energy dimension, or total and per-asset CO_2 emissions for the environmental dimension.

For each selected KPI, we estimate the mean performance from $n$independent simulation replications and summarize the estimation uncertainty using a two-sided 95% Confidence Interval (CI) [93]. Let $y_1, \ldots, y_n$ denote the KPI values obtained from the replications, $\bar{y}$ their sample mean, and s their sample standard deviation. The 95% CI for the mean is then computed as

$$\bar{y} \pm t_{0.975,n-1}\frac{s}{\sqrt{n}},$$

where $t_{0.975,n-1}$ is the critical value of the Student's t-distribution with $n-1$ degrees of freedom. Behavioral agreement is assessed for the intended use by comparing the corresponding point estimates and 95% CIs from the extracted and reference models. Close point estimates and overlapping CIs are interpreted as supporting behavioral agreement for the selected KPIs and the considered experimental setting. Because the reference system in this study is an illustrative simulation model rather than an industrial manufacturing system, this assessment supports proof-of-concept feasibility and behavioral consistency but does not establish full industrial validity.

3.5. Comprehensive Digital Twin Framework

A CDT extends conventional DTs by expanding the modeled system representation from a primarily time-centered view to a multidimensional perspective. In the workflow shown in Fig. 7, data streams from IoT devices and controllers are collected, cleaned, and integrated into a multidimensional event log that captures the relevant entities and performance dimensions. Based on this log, MFPM discovers dimension-specific process flows that are integrated into a unified MDSPN, which serves as the CDT's executable simulation core and supports continuous validation and updating against observed system behavior. The validated CDT is then used for what-if analysis and KPI-based multi-objective decision support in SMSs. We assess the extracted CDT simulation model through two complementary validation activities: (i) structural validation and (ii) operational or output validation. Structural validation assesses whether the discovered MDSPN structure and logic are reasonable with respect to the real system; accordingly, domain experts review the extracted places, transitions, arcs, and firing logic to confirm that they reproduce the observed flow relations. For the illustrative case study, where a reference (ground-truth) model is available, this step is supported by a direct structural comparison. Operational validity is assessed by comparing the real-system and simulated KPIs under the same experimental conditions; agreement is indicated when the corresponding 95% CIs overlap. In this study, we demonstrate how the extracted CDT, after showing structural consistency and operational

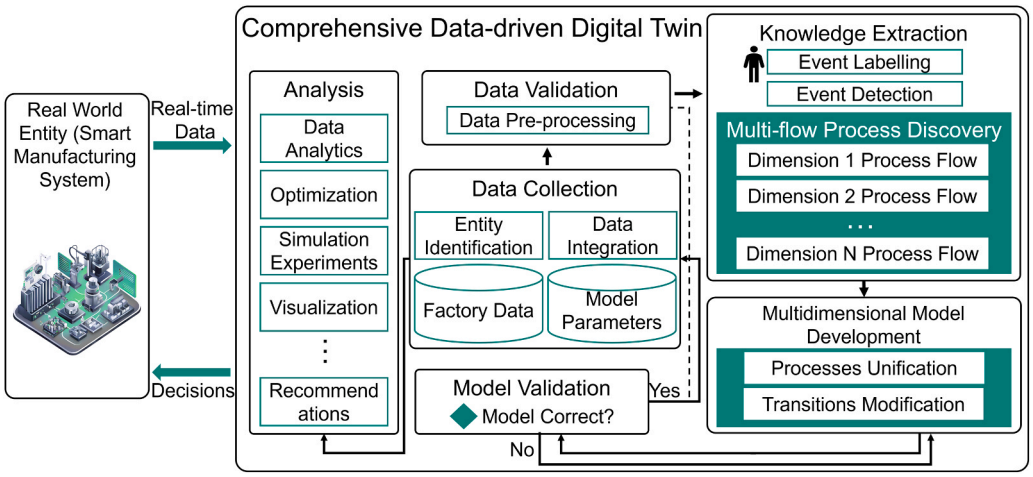


Fig. 7. Framework for comprehensive data-driven Digital Twins of smart manufacturing systems [21].

agreement for the selected KPIs and experimental setting, can be used as an executable model for proof-of-concept simulation-based multi-objective decision support.

3.6. *Integration of a Multi-Objective Evolutionary Algorithm into the Comprehensive Digital Twin*

For simulation-based multi-objective decision support, the optimization-ready CDT is integrated with an MOEA. In this integration, the MDSPN model M serves as the executable evaluation model, while the MOEA explores alternative operating configurations by repeatedly configuring, simulating, and evaluating M. Each candidate configuration is represented by a decision vector $x = (x_1, \ldots, x_n) \in \mathscr{X}$, where $\mathscr{X}$ denotes the admissible decision space. For continuous decision variables, this space can be expressed as $\mathscr{X} = \{x | l \le x \le u\}$, where l and u are the lower and upper bounds defined during optimization preparation. The optimization-preparation layer establishes the correspondence between the elements of x and the controllable parameters represented in the MDPNML model. For each candidate x, the corresponding parameter values are applied to a model instance while all non-controllable model elements remain unchanged. The resulting configured model is denoted by $M(x)$.

Because $M(x)$ is stochastic, its performance is estimated by executing r independent simulation replications over a fixed simulation horizon H. Let $K^{(j)}(x)$ denote the vector of KPIs observed in replication j. The estimated KPI vector is then

$$\widehat{K}(x) = \frac{1}{r} \sum_{j=1}^{r} K^{(j)}(x).$$

The estimated KPIs are transformed into the objective vector $\widehat{F}(x) = \Psi(\widehat{K}(x), \widehat{K}_{\text{base}})$, where Ψ denotes the application-specific objective definition. The transformation may include objective selection, minimization or maximization conventions, and normalization relative to a baseline configuration. The baseline estimate $\widehat{K}_{\text{base}}$ is obtained by simulating a baseline configuration x_{base} before the evolutionary search and is used for performance comparison and result interpretation.

Fig. 8 presents the complete CDT–MOEA workflow. The optimization problem specifies the admissible decision space and simulation settings. The selected multi-objective evolutionary algorithm is defined by a population size N, mating-selection and variation operators, an environmental-selection mechanism, and a termination criterion T. The termination criterion may be based on a generation limit, an evaluation budget, a computational-time budget, or another application-appropriate stopping rule.

The procedure begins by evaluating the baseline configuration and

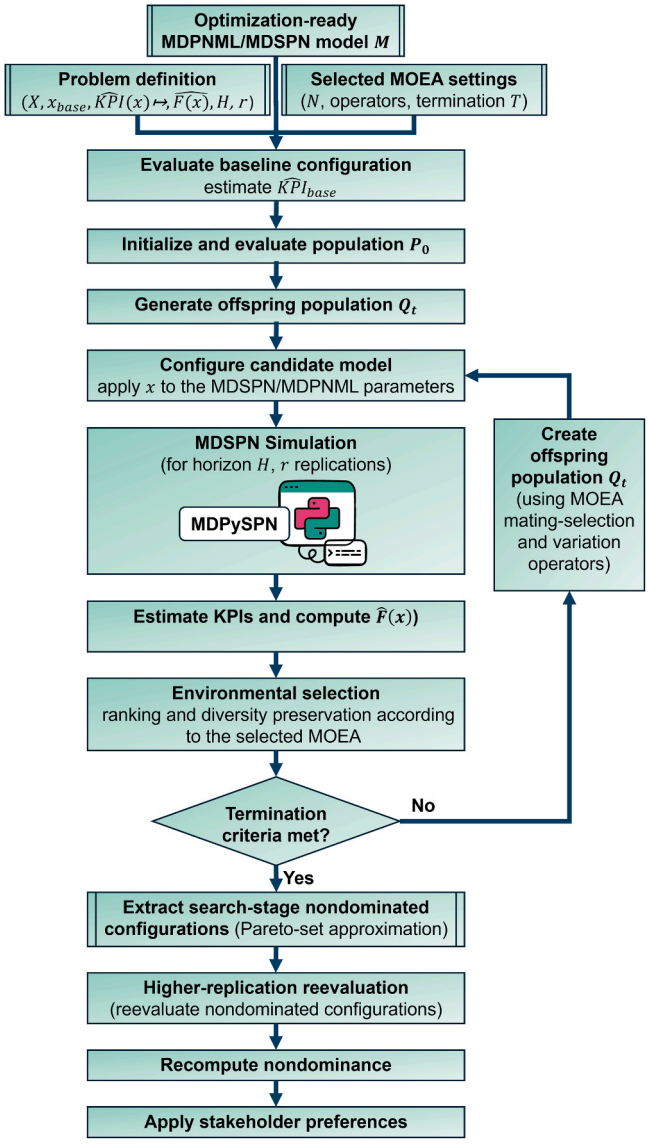


Fig. 8. Workflow for integrating the MDSPN-based CDT with a multi-objective evolutionary optimizer in an optimization-via-simulation loop.

generating an initial population P_0. Every candidate in P_0 is evaluated using the same CDT-based evaluation procedure. The MOEA then generates an offspring population Q_t from the current parent population P_t. Each offspring configuration is applied to the corresponding MDPNML parameters, evaluated through repeated MDSPN simulation, and transformed into an estimated objective vector. Environmental selection subsequently constructs the next population according to the ranking and diversity-preservation mechanisms of the selected MOEA. This process continues until the termination criterion is satisfied.

At the end of the evolutionary search, the nondominated decision vectors form a finite approximation of the Pareto-optimal set. This Pareto set approximation contains the operating configurations in the decision space. The corresponding objective vectors form the Pareto-front approximation in the objective space and represent the estimated performance of those configurations. Because these objective values are estimated using stochastic simulation, the search-stage nondominated configurations are reevaluated using a larger number of independent replications to obtain more precise objective estimates [58]. The decision vectors remain unchanged during this reevaluation. Nondominance is subsequently recomputed from the updated estimates because some search-stage nondominated configurations may become dominated after higher-replication reevaluation. Stakeholder preferences are applied only to the resulting reevaluated nondominated configurations. This enables trade-off exploration without predefined priorities and subsequent scenario selection according to stakeholder preferences.

The search is limited to the parameter ranges defined during optimization preparation and for which the model structure and parameterization remain applicable. Configurations outside these ranges, or involving behavior not represented in the MDSPN, require further assessment and, where necessary, revalidation. If simulation-based operational constraints are included, the evaluation may additionally return an estimated constraint vector $\widehat{g}(x)$. Such constraints can then be handled using the feasibility and constraint-handling mechanism of the selected MOEA.

Algorithm 1 formalizes the CDT–MOEA workflow presented in Fig. 8. The algorithm requires the optimization-ready MDSPN model $M$and associated optimization metadata, the MOEA settings, and the stakeholder preference scenarios. The optimization metadata defines the admissible decision space $\mathscr{X}$, the mapping between decision variables and model parameters, the baseline configuration x_{base}, the KPI and objective definitions, the simulation horizon H, and the replication counts r_{base}, r, and r_{final}. The MOEA settings define the population size N, the mating-selection and variation operators, the environmental-selection mechanism, and the termination criterion T.

The CDT evaluation procedure receives a candidate configuration $x$and a replication count R. In substep (i), an independent instance of M is created for each replication, the values of $x$are applied to the corresponding controllable parameters, the configured model is simulated over horizon H, and the resulting KPI vector $K^{(j)}(x)$ is recorded. Substep (ii) computes the estimated KPI vector $\widehat{K}(x)$and the corresponding objective vector $\widehat{F}(x)$. Substep (iii) returns both estimates.

The optimization procedure begins in Step 1 by simulating the baseline configuration x_{base}using r_{base}independent replications and computing $\widehat{K}_{\text{base}}$. Step 2 generates the initial population P_0 from $\mathscr{X}$ and evaluates each candidate using the CDT evaluation procedure with $r$replications. Step 3 initializes the generation counter t. In Step 4, the selected MOEA repeatedly generates the offspring population Q_t, evaluates the offspring configurations, applies environmental selection to construct P_{t+1}, and updates t until the termination criterion T is satisfied. Step 5 extracts the search-stage nondominated set X_{ND}. Step 6 reevaluates the configurations in X_{ND} using r_{final} replications and recomputes nondominance to obtain $X^\star$. Step 7 applies the stakeholder preference scenarios and returns $X^\star$, the corresponding objective values $\widehat{F}(X^\star)$, the KPI estimates, and the selected operating scenarios.

Algorithm 1. Integration of a Multi-Objective Evolutionary Optimizer into the Comprehensive Digital Twin.

Input: Optimization-ready MDSPN-based CDT model M and associated optimization metadata; MOEA settings, including the termination criterion T; and stakeholder preference scenarios.

Output: Reevaluated nondominated set $X^\star$, corresponding objective values $\hat{F}(X^\star)$, associated KPI estimates, and stakeholder-selected operating scenarios.

CDT evaluation procedure:
For a candidate configuration x and replication count R:

i. For each replication $j = 1, \ldots, R$:
   a. Create an independent instance of the MDSPN model M.
   b. Apply the elements of x to the corresponding controllable model parameters.
   c. Simulate the configured model over horizon H.
   d. Extract the resulting KPI vector $K^{(j)}(x)$.

ii. Compute: $\widehat{K}(x) = \frac{1}{R}\sum_{j=1}^{R} K^{(j)}(x)$ *and* $\hat{F}(x) = \Psi\left(\widehat{K}(x), \widehat{K}_{\text{base}}\right)$.

iii. Return $\widehat{K}(x)$and $\hat{F}(x)$.

Optimization procedure:

1. Simulate $M(x_{\text{base}})$ for r_{base} independent replications and compute the baseline KPI estimate $\widehat{K}_{\text{base}}$.
2. Generate $P_0 = \{x^{(1)}, \ldots, x^{(N)}\} \subseteq \mathcal{X}$ and evaluate every $x \in P_0$ using Evaluate(x, r).
3. Set $t \leftarrow 0$.
4. While T is not satisfied:
   a. Generate Q_t from P_t using the mating-selection and variation operators.
   b. Evaluate every $x \in Q_t$ using Evaluate(x, r).
   c. Construct P_{t+1} using the environmental-selection mechanism of the selected MOEA.
   d. Set $t \leftarrow t + 1$.
5. Determine the search-stage nondominated subset: $X_{\text{ND}} \subseteq P_t$.
6. Reevaluate every $x \in X_{\text{ND}}$ using Evaluate(x, r_{final}), and determine the reevaluated nondominated set: $X^\star \subseteq X_{\text{ND}}$.
7. Apply the stakeholder preference scenarios to $X^\star$, identify the corresponding operating scenarios, and return $X^\star$, $\hat{F}(X^\star)$, the associated KPI estimates, and the selected operating scenarios.

In this study, Algorithm 1 is instantiated with NSGA-II. In Step 4, the parent and offspring populations are merged as $R_t = P_t \cup Q_t$, ranked using fast nondominated sorting, and filtered using crowding distance to preserve diversity. The next population is formed by admitting fronts in

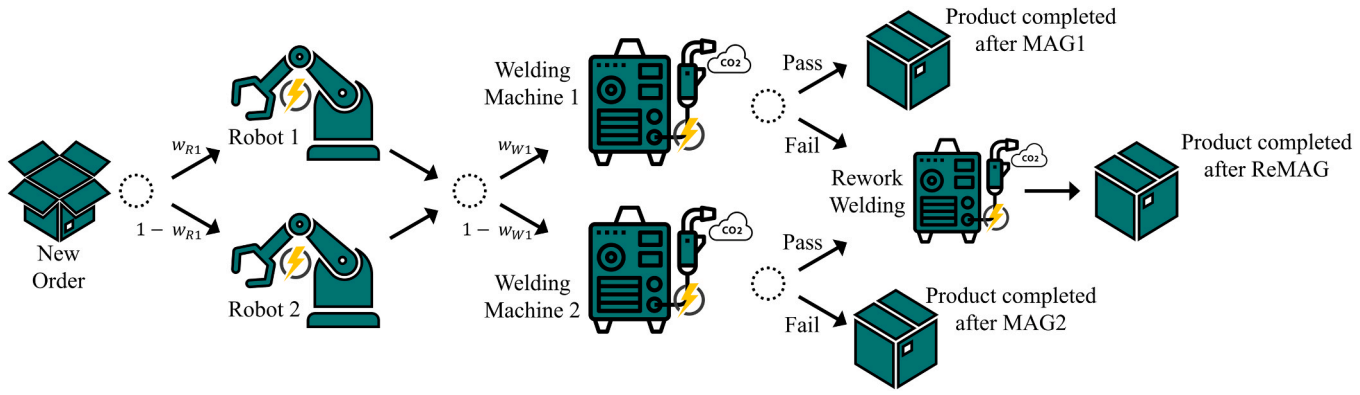


Fig. 9. Schematic layout of the illustrative manufacturing line.

rank order and, when truncation is required, retaining solutions with the largest crowding distances [31].

3.7. Integration of Weighted-Sum Scalarization into the Comprehensive Digital Twin

To provide an alternative optimization formulation using the same optimization-ready CDT, the MDSPN–MDPNML evaluation pipeline can also be integrated with weighted-sum scalarization. Weighted-sum scalarization transforms a multi-objective optimization problem into a single-objective problem by assigning weights to the individual objectives and aggregating the weighted objective values into a scalar objective function [59,62]. In contrast to the Pareto-based approach described in Section 3.6, which preserves the objectives separately during the search and introduces stakeholder preferences after obtaining a Pareto-front approximation, weighted-sum scalarization requires the relative objective weights to be specified before optimization. The resulting search therefore identifies a preference-dependent operating configuration rather than a set of alternative trade-off configurations [59,60,62].

The scalarized workflow uses the same optimization-ready MDSPN model M, admissible decision space $\mathscr{X}$, decision-to-model mapping, baseline configuration *x_base*, and stochastic CDT evaluation procedure defined in Section 3.6. For a candidate decision vector $x \in \mathscr{X}$, the optimization-preparation layer applies the elements of x to the corresponding controllable MDPNML parameters and creates the configured model $M(x)$, while all non-controllable model elements remain unchanged. The configured model is evaluated using r independent simulation replications over the simulation horizon H. The estimated KPI vector is therefore calculated as

$$\widehat{K}(x) = \frac{1}{r}\sum_{j=1}^{r} K^{(j)}(x),$$

where $K\hat{}(j)(x)$ denotes the KPI vector obtained from replication j. The same application-specific objective transformation used in the Pareto-based workflow is then retained:

$$\widehat{F}(x) = \Psi(\widehat{K}(x), \widehat{K}_{\text{base}}) = (\widehat{f}_1(x), \ldots, \widehat{f}_m(x)),$$

where Ψ defines objective selection, minimization or maximization conventions, and any required baseline scaling, while $\widehat{K}_{base}$ denotes the baseline KPI estimate. Retaining the same transformation ensures that the Pareto-based and scalarized formulations operate on the same simulation-derived objective criteria. Let $\lambda = (\lambda_1, \ldots, \lambda_m)$ denote the vector of a priori objective weights, with

$$\lambda_i \geq 0, \sum_{i=1}^{m} \lambda_i = 1.$$

The scalarized objective estimate is defined as

$$\widehat{J}(x;\lambda) = \sum_{i=1}^{m} \lambda_i \widehat{f}_i(x) = \lambda^T \widehat{F}(x).$$

The corresponding simulation-based single-objective optimization problem seeks

$$x^*_\lambda = \arg\min_{x \in X} \widehat{J}(x;\lambda)$$

The scalarized optimization workflow therefore follows the same CDT evaluation path as Algorithm 1 through calculation of $\widehat{F}(x)$. The workflows diverge after the objective vector has been obtained. Algorithm 1 retains the objective components separately and applies the environmental-selection mechanism of the selected MOEA. The scalarized workflow instead calculates $\widehat{J}(x;\lambda)$ and ranks candidate configurations according to this single scalar value. A single-objective optimizer can then generate and select candidate configurations within $\mathscr{X}$ using the scalarized objective as the optimization criterion.

Table 2
Case study activities, modes, and dimensional features.

Activity	Mode	Time distribution (min)	Energy-consumption rate (kW)	CO_2-emission rate (g/min)
Order arrivals	-	Expon (1/5.5, 1)	-	-
Robot 1	processing	Triang (9, 9.5, 10)	0.40	-
	idle	-	0.08	-
Robot 2	processing	Triang(6, 6.5, 7)	0.75	-
	idle	-	0.15	-
Welding Machine 1 (MAG 1)	processing	Triang (9,9.5,10)	4.00	5.00
	idle	-	0.30	-
Welding Machine 2 (MAG 2)	processing	Triang (7,7.5,8)	6.50	2.40
	idle	-	0.45	-
Rework Welding Machine	processing	Triang (11| 11.5| 12)	3.20	3.50
	idle	-	0.35	
Rework Welding Machine	restart	Det (1)	Processing energy rate	-
*All (Except for ReMAG)	restart	Det (0.5)	Processing (energy rate *2)	-

Because $\widehat{J}(x;\lambda)$ is estimated from stochastic simulation outputs, sampling variability can affect the ranking of candidate configurations [58]. After the search terminates, the best search-stage configuration is therefore reevaluated using r_{final} independent replications. The baseline configuration is reevaluated using the same higher-replication evaluation stage, and $\widehat{F}(x)$ and $\widehat{J}(x;\lambda)$ are recomputed from the updated KPI estimates. The decision vector remains unchanged during this reevaluation.

Unlike the Pareto-based workflow, the scalarized workflow does not require extraction of a nondominated set, recomputation of nondominance, or post-search preference selection because the preference information is already encoded in λ. Changing the objective weights represents a different a priori preference and generally requires a new scalarized optimization run [59–62].

Algorithm 2 uses the same CDT evaluation procedure as Algorithm 1 through the calculation of the estimated KPI vector $\widehat{K}(x)$ and the corresponding objective vector $\widehat{F}(x)$. The scalarized evaluation procedure then calculates the scalarized objective $\widehat{J}(x;\lambda)$ from the objective vector and the predefined weight vector λ, and returns the KPI estimate, objective vector, and scalarized objective value to the single-objective optimizer. The scalarized optimization procedure begins in Step 1 by simulating the baseline configuration x_{base} using r_{base} independent replications and computing $\widehat{K}_{base}$. Step 2 generates the initial population P_0 from $\mathscr{X}$ and evaluates each candidate using the scalarized CDT evaluation procedure with r replications. Step 3 initializes the generation counter t. In Step 4, the selected single-objective optimizer repeatedly generates the offspring population Q_t, evaluates the offspring configurations, constructs P_{t+1} according to the scalarized objective $\widehat{J}(x;\lambda)$ and the optimizer's selection mechanism, and updates t until the termination criterion T is satisfied. Step 5 identifies the best search-stage scalarized configuration $x^{search}{}_{\lambda}$ as the configuration with the minimum scalarized objective value in the final population. Step 6 reevaluates the baseline configuration x_{base} and $x^{search}{}_{\lambda}$ using r_{final} independent replications and recomputes the higher-replication baseline KPI estimate, the objective vector $\widehat{F}(x^{search}{}_{\lambda})$, and the scalarized objective $\widehat{J}(x^{search}{}_{\lambda};\lambda)$. The decision vector remains unchanged during this reevaluation. Step 7 sets $x^{*}{}_{\lambda}$ equal to $x^{search}{}_{\lambda}$ and returns $x^{*}{}_{\lambda}$, the corresponding objective vector $\widehat{F}(x^{*}{}_{\lambda})$, the scalarized objective value $\widehat{J}(x^{*}{}_{\lambda};\lambda)$, and the associated KPI estimates.

Algorithm 2. Integration of Weighted-Sum Scalarization into the Comprehensive Digital Twin.

Input: Optimization-ready MDSPN-based CDT model M and associated optimization metadata; scalarization weights λ; and single-objective optimizer settings, including termination criterion T.

Output: Reevaluated scalarized solution $x_{\lambda}^{\star}$, corresponding objective vector $\hat{F}(x_{\lambda}^{\star})$, scalarized objective value $\hat{J}(x_{\lambda}^{\star};\lambda)$, and associated KPI estimates.

CDT evaluation procedure:
For a candidate configuration x and replication count R:

i. For each replication $j = 1, \ldots, R$:
   a. Create an independent instance of the MDSPN model M.
   b. Apply the elements of x to the corresponding controllable model parameters.
   c. Simulate the configured model over horizon H.
   d. Extract the resulting KPI vector $K^{(j)}(x)$.

ii. Compute: $\hat{K}(x) = \frac{1}{R}\sum_{j=1}^{R} K^{(j)}(x)$ *and* $\hat{F}(x) = \Psi\left(\hat{K}(x), \hat{K}_{\text{base}}\right)$.

iii. Compute the scalarized objective $\hat{J}(x;\lambda) = \lambda^{T}\hat{F}(x)$.

iv. Return $\hat{K}(x)$, $\hat{F}(x)$ and $\hat{J}(x;\lambda)$.

Optimization procedure:

1. Simulate $M(x_{\text{base}})$ for r_{base} independent replications and compute the baseline KPI estimate $\hat{K}_{\text{base}}$.
2. Generate $P_0 = \{x^{(1)}, \ldots, x^{(N)}\} \subseteq \mathcal{X}$ and evaluate every $x \in P_0$ using EvaluateScalarized(x, r).
3. Set $t \leftarrow 0$.
4. While T is not satisfied:
   a. Generate Q_t from P_t using the mating-selection and variation operators of the selected single-objective optimizer.
   b. Evaluate every $x \in Q_t$ using EvaluateScalarized(x, r).
   c. Construct P_{t+1} according to the scalarized objective $\hat{J}(x;\lambda)$ and the selection mechanism of the selected single-objective optimizer.
   d. Set $t \leftarrow t + 1$.
5. Determine the best search-stage scalarized configuration: $x_{\lambda}^{search} = arg\ min_{x \in P_t} \hat{J}(x;\lambda)$.
6. Reevaluate x_base and x_{λ}^{search} using r_{final} independent replications; recompute the higher-replication baseline estimate, the objective vector $\hat{F}\left(x_{\lambda}^{search}\right)$, and the scalarized objective $\hat{J}\left(x_{\lambda}^{search};\lambda\right)$.
7. Set $x_\lambda\text{^}\star \leftarrow x_{\lambda}^{search}$ and return $x_\lambda\text{^}\star$, $\hat{F}(x_{\lambda}^{\star})$, $\hat{J}(x_{\lambda}^{\star};\lambda)$, and the associated KPI estimates.

4. Illustrative Case Study

In this section, we present an illustrative manufacturing case study to demonstrate the proposed CDT-based multi-objective decision-support workflow. We implemented a discrete-event simulation model of the manufacturing line in AnyLogic [94] and used it as the reference model to generate multidimensional event-log data. We then applied MFPM, supported by data analysis, to extract the corresponding MDSPN-based CDT from the event logs. Next, we assessed the extracted model through structural and operational validation, prepared it for optimization, and integrated it into the optimization-via-simulation workflow introduced in Section 3. For the case study, we instantiated the Pareto-based workflow with NSGA-II and evaluated weighted-sum scalarization as a comparative baseline, using MDPySPN to execute and evaluate candidate MDSPN configurations. We formulated three objectives: minimizing grid-energy consumption, minimizing CO_2 emissions, and maximizing the number of completed products. For the Pareto-based workflow, rather than selecting a single universally preferred configuration, we generated a set of nondominated configurations representing different trade-offs among the three objectives. We subsequently reevaluated each search-stage nondominated configuration using r_{final} independent simulation replications and recomputed nondominance using the updated objective estimates. Finally, we applied one post-search preference rule and evaluated the selected scenario in the reference model.

Table 3
Excerpt of the generated multidimensional event log for the production flow.

Time Stamp	ID	Asset	Energy Stamp (kW.min)	Energy Type	Emission Stamp (g)	Emission Type	Event
00:01:24	267	Robot 2	31.650	Grid	0.0	NA	Robot 2 End
00:01:32	267	MAG2	359.134	Grid	126.142	CO_2	MAG 2 End
00:01:32	267	NA	0.0	NA	0.0	NA	MAG 2 Failed
00:01:44	267	ReMAG	44.249	Grid	37.569	CO_2	ReMAG End

Table 4
Excerpt from the generated asset-specific multidimensional event log for Welding Machine 1.

Time Stamp	ID	Asset	Energy Stamp (kW.min)	Energy Type	Emission Stamp (g)	Emission Type	Event
00:01:40	3	MAG 1	122.748	Grid	0.000	NA	Idle End
00:01:54	3	MAG 1	126.748	Grid	0.000	NA	Restart End
00:01:54	3	MAG 1	126.748	Grid	137.810	CO_2	Active Begin
00:02:32	3	MAG 1	277.596	Grid	326.370	CO_2	Active End

4.1. Case Study Description

As we illustrate in Fig. 9, the illustrative manufacturing line consists of two parallel robotic resources, Robot 1 and Robot 2, followed by two parallel primary welding resources, Welding Machine 1 (MAG1) and Welding Machine 2 (MAG2). A shared Rework Welding Machine (ReMAG) processes jobs that fail after the primary welding operation. Jobs arrive according to a shifted exponential distribution with $\lambda_A = 1/5.5$ arrivals per minute and a minimum interarrival time of 1 min, resulting in an expected interarrival time of 6.5 min. The routing logic assigns each arriving job first to one of the two robots and then to one of the two primary welding machines. In the reference model, $w_{R1} = 0.5$ and $w_{W1} = 0.5$, while $1 - w_{R1}$ and $1 - w_{W1}$ define the routing weights for Robot 2 and MAG2, respectively. Jobs processed by MAG1 and MAG2 pass with probabilities of 0.95 and 0.85. Passed jobs leave the system as completed products, while failed jobs undergo an additional operation at ReMAG before completion. The number of completed products includes jobs completed after primary welding or rework.

Although the paired resources perform the same production functions, they differ in processing times, energy-consumption rates, and CO_2-emission rates. Processing times follow triangular distributions parameterized by minimum, mode, and maximum values. Robot 1 and MAG1 follow Triang $(9, 9.5, 10)$, Robot 2 follows Triang $(6, 6.5, 7)$, MAG2 follows Triang$(7, 7.5, 8)$, and ReMAG follows Triang $(11, 11.5, 12)$, with all parameters expressed in minutes. During processing, Robot 1, Robot 2, MAG1, MAG2, and ReMAG consume energy at rates of 0.40, 0.75, 4.00, 6.50, and 3.20 kW, respectively; the corresponding idle rates are 0.08, 0.15, 0.30, 0.45, and 0.35 kW. The model associates CO_2 emissions with the welding operations at rates of 5.00 g/min for MAG1, 2.40 g/min for MAG2, and 3.50 g/min for ReMAG. Table 2 summarizes the processing-time distributions, operating modes, energy-consumption rates, and CO_2-emission rates used in the reference model.

Each asset has four operating states: processing, idle, shutdown, and restart. After completing an operation, the asset enters the idle state and shuts down when its continuous idle time reaches the corresponding threshold τ. The reference model uses a threshold of 5 min for all assets. If a job is assigned to an asset in the shutdown state, the restart operation begins, followed by processing.

The five idle-to-shutdown thresholds and two routing weights form the decision vector $x = (\tau_{R1}, \tau_{R2}, \tau_{W1}, \tau_{W2}, \tau_{RW}, w_{R1}, w_{W1})$. Threshold variables correspond to Robot 1, Robot 2, MAG1, MAG2, and Rework Welding. The routing weights w_{R1} and w_{W1} define selection of Robot 1 and MAG1, while the complementary weights apply to Robot 2 and MAG2. The arrival process, processing-time distributions, quality probabilities, restart settings, and dimensional rates remain fixed.

4.2. Multidimensional Event-Log Generation

The illustrative manufacturing line was simulated for 100 independent daily replications, each representing a single 8-h production shift from 08:00–16:00 and using a different random seed. Each replication generated a multidimensional event log for the overall production flow and a separate multidimensional event log for each asset. The production-flow log recorded order arrivals, routing decisions, processing events, and product-completion events, whereas the asset-specific logs recorded transitions between operating states together with the corresponding energy-consumption and CO_2-emission information. Table 3 presents an excerpt from the production-flow log, and Table 4 presents an excerpt from the asset-specific log of MAG1. Event-log timestamps are reported as elapsed time from the beginning of each 8-h shift. At the end of each replication, system-level and asset-level KPIs were also recorded, including the number of arriving orders, the number of completed products, the throughput (calculated as the number of completed products divided by the number of arriving orders), total and asset-level grid-energy consumption in kWh, and total and asset-level CO_2 emissions in g. The Energy Stamp values in Tables 3 and 4 represent cumulative energy in kW·min. For reporting the energy KPIs in the result tables and figures, these values were converted to kWh by dividing by 60.

Table 5
Excerpt of the extracted multidimensional transition table.

#	Transition	Time	Distribution	Weight	Energy	Energy Type	Energy Value	Emission	Emission Type	Emission Value
1	New order arrival	C	Expon (λ=0.188 min−1, shift=1.010 min)	-	-	-	-	-	-	-
2	Select Robot1	-	-	0.511	-	-	-	-	-	-
3	Robot1 processing	C	Triang (8.99, 9.55, 9.98)	-	C	Rate	0.4	-	-	-
4	Select MAG1	-	-	0.491	-	-	-	-	-	-
5	MAG1 processing	C	Triang (8.99, 9.55, 10.00)	-	C	Rate	4	C	Rate	5
…	…	…	…	…	…	…	…	…	…	…

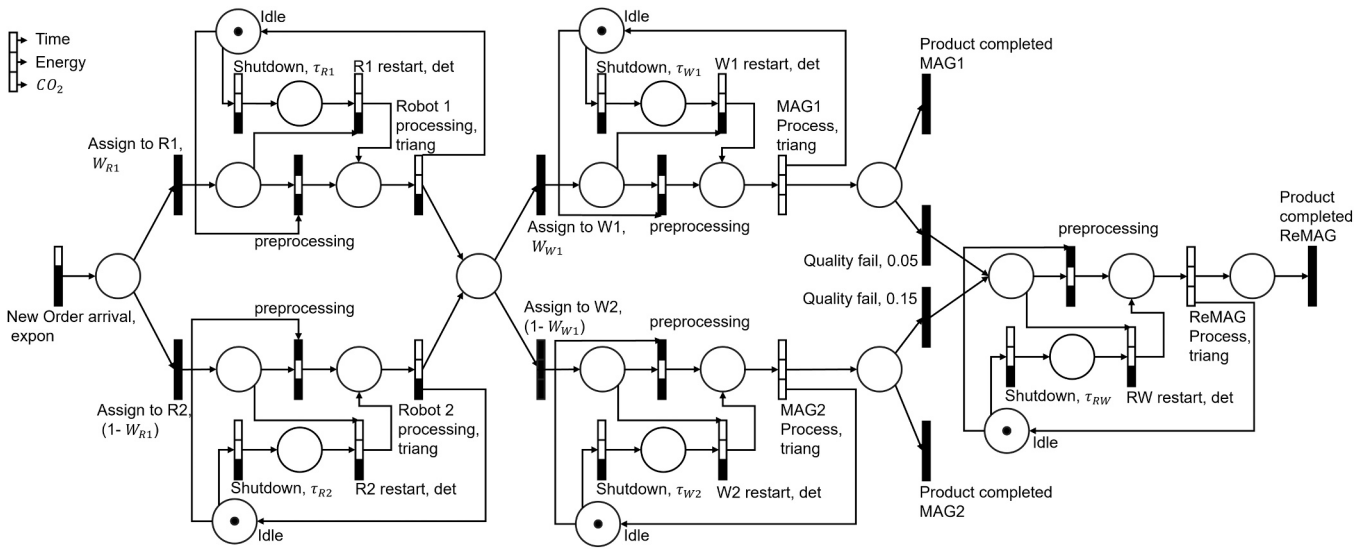


Fig. 10. Multidimensional Stochastic Petri Net extracted from the multidimensional event logs of the illustrative manufacturing line.

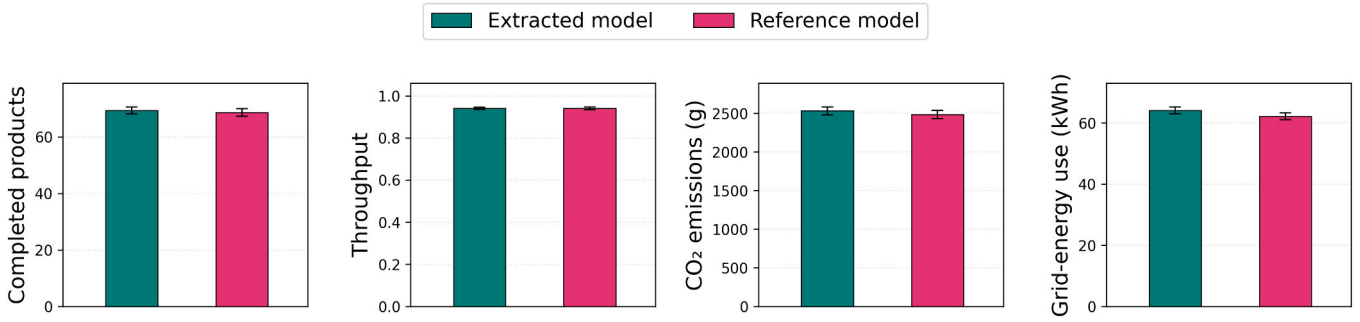


Fig. 11. Comparison of the extracted MDSPN and the reference model based on mean KPI values and 95% confidence intervals over 100 independent 8-h replications.

Table 6
Decision variables and bounds for the decision vector x.

Variable	Description	Baseline value	Range
τ_{R1}, τ_{R2}	Idle-to-shutdown thresholds for Robot 1/2	5 min	[0, 20]
$\tau_{W1}, \tau_{W2}, \tau_{RW}$	Idle-to-shutdown thresholds for MAG 1,2 and ReMAG	5 min	[0, 20]
w_{R1}	Routing weight for Robot 1 (Robot 2 is $(1 - w_{R1})$)	0.511	[0, 1]
w_{W1}	Routing weight for MAG 1 (MAG 2 is $(1 - w_{W1})$)	0.491	[0, 1]

4.3. CDT Model Extraction, Simulation and Validation

Using the generated multidimensional event logs, the MDSPN-based CDT was extracted following the automated MDSPN extraction procedure described in Section 3.3. We combined the logs from ten daily 8-h production shifts to rediscover the production-flow structure and estimate the stochastic and multidimensional parameters. Following the MFPM-based workflow, we processed the cleaned production-flow and asset-specific logs to extract places, transitions, arcs, routing behavior, processing-time distributions, and dependencies among asset operating states. We organized the extracted information in a Multidimensional Transition Table, which served as input for constructing the MDSPN. Table 5 presents a representative excerpt from the extracted Multidimensional Transition Table. The estimated shifted-exponential arrival distribution had a shift of 1.010 min and a rate of approximately 0.188 min^{-1}, closely reproducing the reference process. The extracted routing weights were 0.511 for Robot 1 and 0.491 for MAG1. The procedure also recovered the processing-time distributions and dimensional rates. For example, Robot 1 followed a triangular distribution with minimum, mode, and maximum processing times of 8.99, 9.55, and 9.98 min, respectively, and an energy-consumption rate of 0.40 kW during processing.

Here, C denotes a transition's contribution to the corresponding dimension. The extracted weights closely matched the reference-model parameters, with only minor deviations resulting from estimation using a finite stochastic event-log sample. The asset-state submodels were connected to the production flow to synchronize resource availability with job progression. The idle-to-shutdown transitions used the extracted baseline duration of 5 min, while idle energy consumption was represented as a rate-based dimensional contribution. Restart durations and energy-consumption rates were incorporated similarly. We redrew the extracted MDSPN in Fig. 10 to improve readability. The structural validation confirmed that the extracted model represents the reference production sequence, the two robot-routing branches, the two primary-welding branches, the quality-outcome branches, the rework path, and the processing–idle–shutdown–restart state relations for all five assets.

We represented the extracted MDSPN in MDPNML and executed it using MDPySPN. To assess behavioral agreement, we evaluated the extracted and reference models using 100 independent 8-h simulation replications. We applied identical KPI definitions and compared the sample means and 95% CIs for the number of completed products,

throughput, total CO_2 emissions, and grid-energy consumption. As shown in Fig. 11, the two models produced similar KPI estimates. The extracted and reference models yielded, respectively, 69.37 and 68.61 completed products, 0.9399 and 0.9404 throughput, 2530.75 and 2482.50 g of CO_2, and 64.05 and 62.14 kWh of grid-energy consumption. These results correspond to relative differences of approximately 1.11%, 0.05%, 1.94%, and 3.08%, respectively. These results support the use of the extracted model in the subsequent proof-of-concept optimization study.

4.4. Case-Study Optimization Formulation and Optimization-Method Instantiation

To prepare the extracted CDT for optimization, we defined the decision vector $x = (\tau_{R1}, \tau_{R2}, \tau_{W1}, \tau_{W2}, \tau_{RW}, w_{R1}, w_{W1})$ and mapped each variable to the corresponding MDPNML element. Table 6 summarizes the baseline values and admissible bounds. All variables are continuous, yielding $\mathscr{X} = [0, 20]^5 \times [0, 1]^2$. Queue capacities and queue-related dispatching logic remained fixed because they were not included in the decision vector.

The baseline configuration of the extracted model is therefore $x_{\text{base}} = (5,\ 5,\ 5,\ 5,\ 5,\ 0.511,\ 0.491)$. The routing weights in x_{base} are the values estimated from the event logs (0.511 and 0.491), rather than the 0.5/0.5 routing probabilities specified in the reference model. Consequently, optimization-stage relative changes are measured against the extracted-model baseline and may reflect the finite-sample estimation error in these parameters.

For each candidate x, the optimization-preparation layer updates the five idle-to-shutdown transitions and two immediate routing transitions in a separate model instance $M(x)$. The arrival process, processing-time distributions, quality probabilities, restart settings, queue capacities, and dimensional rates remain fixed. The optimization minimizes total grid-energy consumption and total CO_2 emissions while maximizing the number of completed products. For each candidate x, the KPI estimates are calculated from r independent simulation replications:

$$\widehat{E}(x) = \frac{1}{r}\sum_{j=1}^{r} E^{(j)}(x),\ \widehat{C}(x) = \frac{1}{r}\sum_{j=1}^{r} C^{(j)}(x),\ \text{and}\ \widehat{P}(x) = \frac{1}{r}\sum_{j=1}^{r} P^{(j)}(x)$$

,

where $E^{(j)}(x)$, $C^{(j)}(x)$, and $P^{(j)}(x)$ denote grid-energy consumption in kWh, CO_2 emissions in g, and the number of completed products in replication j, respectively. Both optimization methods used the same baseline-scaled objective components

$$\widehat{F}(x) = \left(\frac{\widehat{E}(x)}{\widehat{E}_{\text{base}}}, \frac{\widehat{C}(x)}{\widehat{C}_{\text{base}}}, -\frac{\widehat{P}(x)}{\widehat{P}_{\text{base}}}\right),$$

while the unscaled KPI estimates were retained for reporting. The negative sign converts the maximization of completed products into the minimization convention used by the optimization formulations. For the Pareto-based formulation, NSGA-II retained the three components of $\widehat{F}(x)$ separately throughout the search, and no objective weights were introduced. For the scalarized formulation, the same three objective components were combined a priori using weighted-sum scalarization. Equal weights were used for the comparative baseline,

$$\lambda_E = \lambda_C = \lambda_P = \frac{1}{3},$$

resulting in the scalar objective

$$\widehat{J}(x) = \frac{1}{3}\frac{\widehat{E}(x)}{\widehat{E}_{base}} + \frac{1}{3}\frac{\widehat{C}(x)}{\widehat{C}_{base}} - \frac{1}{3}\frac{\widehat{P}(x)}{\widehat{P}_{base}}.$$

The equal weighting provides a transparent comparative scalarization setting rather than a generally preferred stakeholder weighting. No additional simulation-based performance constraint was imposed in either search because the number of completed products was represented directly as an objective.

Table 7
Simulation and optimization settings used in the case-study experiments.

Setting	NSGA-II	Weighted-sum scalarization
Simulation horizon	$H = 480$ min	$H = 480$ min
Decision variables	7 continuous	7 continuous
Search-stage replications	$r = 5$	$r = 5$
Higher-replication reevaluation	$r_{final} = 100$	$r_{final} = 100$
Population size	$N = 80$	$N = 80$
Termination	20 generations	20 generations
Objective treatment	Three components of $\widehat{F}(x)$ retained separately	Equal-weight scalar objective $\widehat{J}(x)$
Objective weights	None	$\frac{1}{3}, \frac{1}{3}, \frac{1}{3}$
Optimizer	NSGA-II	Genetic Algorithm (GA)
Initialization	Uniform random, including x_{base}	Uniform random, including x_{base}
Replication seeds	Random	Random
Duplicate elimination	Enabled	Enabled

For both optimization methods, the baseline configuration was evaluated before the search, included once in the initial population, and used for objective scaling, comparison, and interpretation. Each candidate was simulated over $H = 480$ min, corresponding to one 8-h production shift, using $r = 5$ independent search-stage replications. For consistent stochastic comparison, the same replication-seed set was applied to all candidate configurations within each evaluation stage. Both methods used a population size of $N = 80$, terminated after 20 generations, applied duplicate elimination, and initialized the remaining population uniformly within the bounds in Table 6. For the case-study instantiations, the Pareto-based formulation was solved using NSGA-II, whereas the scalarized single-objective problem was solved using a real-valued Genetic Algorithm (GA). GA is a population-based evolutionary search method that iteratively evolves candidate solutions using selection and variation operators, such as crossover and mutation [95]. Both algorithms were implemented in Python using pymoo [96], and MDPySPN was used to configure and simulate the candidate models.

The lower threshold bound of 0 min permits immediate shutdown after an asset becomes idle, whereas the upper bound of 20 min permits longer idle operation before shutdown. Routing-weight bounds of 0 and 1 permit complete allocation to one parallel branch. These endpoint values were retained for the proof-of-concept search because they produce structurally executable configurations in the case-study model. In an industrial system, experts would restrict these bounds when immediate shutdown or complete branch exclusion is technically, operationally, or contractually inadmissible.

The MDPNML preparation, simulation, NSGA-II and weighted-sum scalarization searches, higher-replication reevaluation, and associated optimization results used in the case study are publicly available in the accompanying repository [97]. Table 7 summarizes the simulation and optimization settings.

For the NSGA-II instantiation, the three components of $\widehat{F}(x)$ were retained separately throughout the search. No scalarization or preference weights were applied. At termination of the NSGA-II search, the nondominated configurations in the final population constituted the search-stage Pareto-set approximation. Because their objective values were estimated using $r = 5$ replications, these configurations were subsequently reevaluated using $r_{final} = 100$ independent replications. Nondominance was then recomputed using the updated objective estimates, and stakeholder preferences were introduced only after this higher-replication reevaluation.

For weighted-sum scalarization, the configuration with the lowest search-stage scalarized objective value $\widehat{J}(x)$ in the final population was retained as the search-stage scalarized solution. The same decision

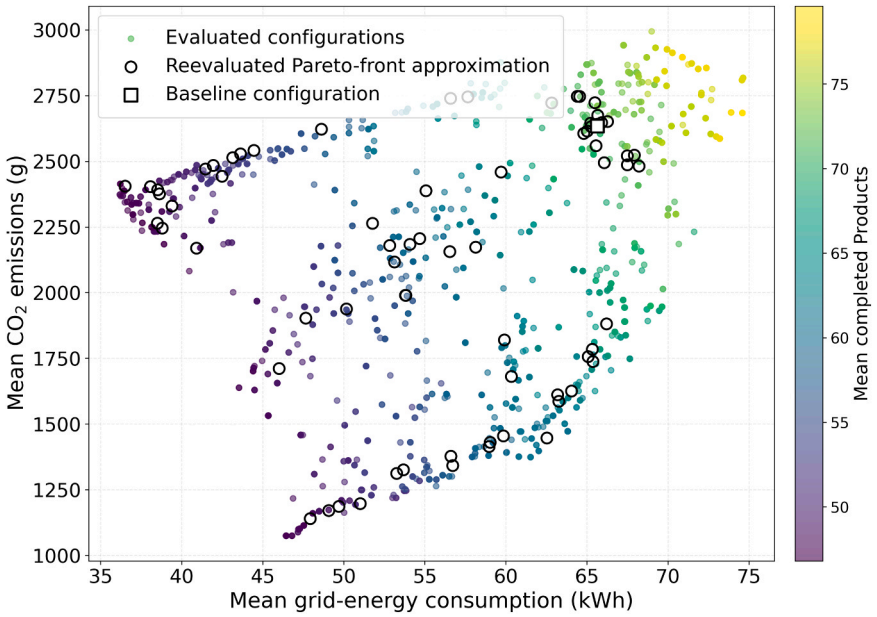


Fig. 12. Search-stage configurations and the reevaluated Pareto-front approximation in the grid-energy–CO_2 objective space.

Table 8
Baseline and objective-extreme configurations after 100-replication reevaluation.

Configuration	Completed products	Throughput	Grid-energy consumption (kWh)	CO_2 emissions (g)	Change in completed products	Change in energy	Change in emissions
Baseline configuration	71.06	0.935	65.64	2633.56	–	–	–
Minimum grid energy	48.07	0.637	36.52	2405.02	-32.35%	-44.37%	-8.68%
Minimum emissions	47.91	0.636	47.94	1139.60	-32.58%	-26.98%	-56.73%
Maximum completed products	72.16	0.945	67.93	2523.09	+1.55%	+3.48%	-4.20%

vector and the baseline configuration were subsequently reevaluated using $r_{final} = 100$ independent replications, and the objective components and scalarized objective were recomputed from the updated estimates. No nondominance recomputation or post-search preference-selection step was required for the scalarized solution because the objective weights were specified before optimization.

4.5. NSGA-II Case-Study Optimization Results

At termination, all 80 configurations in the final population were nondominated according to their five-replication search-stage estimates. The same 80 decision vectors were subsequently reevaluated using 100 independent replications. After recomputing nondominance using the updated objective estimates, 70 configurations remained nondominated, while 10 became dominated. The change in nondominance status illustrates how sampling variability in stochastic objective estimates can affect the Pareto-front approximation and highlights the importance of higher-replication reevaluation before using the Pareto-front approximation for scenario selection. Fig. 12 presents the search-stage configurations in the grid-energy-CO_2 objective space. Marker

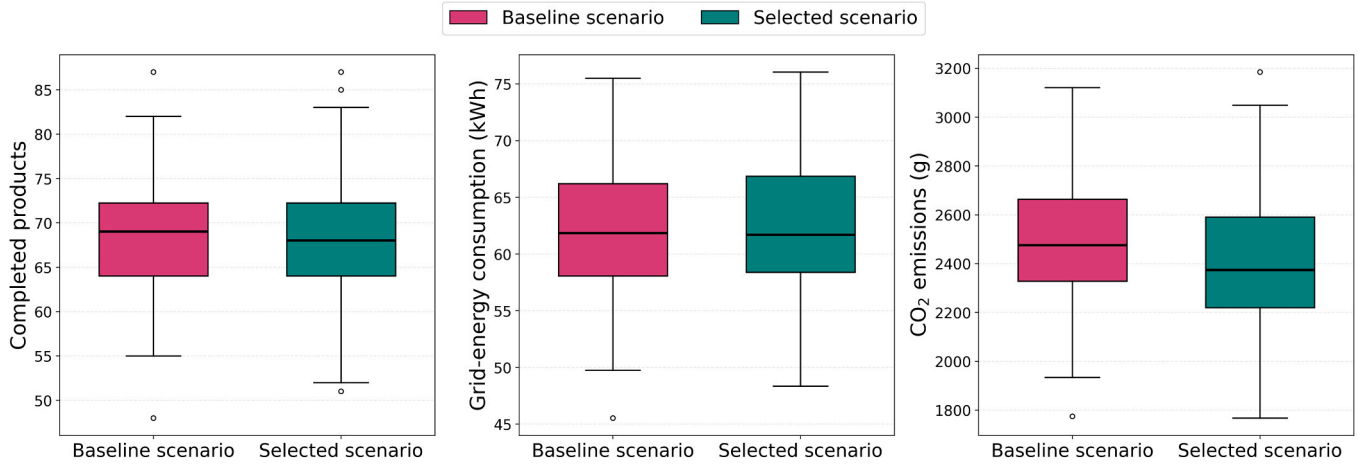


Fig. 13. Distributions of completed products, grid-energy consumption, and CO_2 emissions under the baseline and selected scenarios.

Table 9
Comparison of the baseline and selected scenarios in the reference model over 100 independent replications.

KPI	Baseline scenario, mean [95% CI]	Selected scenario, mean [95% CI]	Change	95% CI for difference
Completed products	68.61 [67.25, 69.97]	68.41 [67.05, 69.77]	−0.29%	[−2.11, 1.71]
Throughput	0.940 [0.934, 0.947]	0.930 [0.921, 0.939]	−1.11%	[−0.0217, 0.0008]
Grid energy consumption (kWh)	62.14 [60.96, 63.31]	62.24 [61.09, 63.39]	+0.17%	[−1.53, 1.74]
CO_2 emissions (g)	2482.50 [2429.18, 2535.82]	2387.96 [2333.88, 2442.04]	−3.81%	[−170.02, −19.07]

color indicates the mean number of completed products, while outlined markers identify the reevaluated nondominated configurations. Because nondominance was determined across all three objectives, some configurations may appear dominated in the two-dimensional projection but remain nondominated because of their completed-product performance. The search-stage markers use five-replication estimates, whereas the outlined markers use 100-replication estimates.

After reevaluation, the nondominated configurations ranged from 36.52 to 68.22 kWh in mean grid-energy consumption, from 1139.60 to 2747.00 g in mean CO_2emissions, and from 47.71 to 72.16 completed products. Their mean throughput ranged from approximately 0.633–0.945. These results reveal trade-offs among completed-product performance, energy consumption, and emissions, reflecting differences between the welding machines in energy-consumption rates, emission rates, quality probabilities, and rework behavior.

The baseline configuration, reevaluated using 100 replications, produced 71.06 completed products, a throughput of 0.935, grid-energy consumption of 65.64 kWh, and CO_2 emissions of 2633.56 g. As shown in Table 8, the minimum-energy configuration reduced energy consumption by 44.37% and emissions by 8.68%, but reduced the number of completed products by 32.35%. The minimum-emission configuration reduced emissions by 56.73% and energy consumption by 26.98%, while reducing completed products by 32.58%. The configuration with the highest number of completed products increased completed products by 1.55% and reduced emissions by 4.20%, but increased energy consumption by 3.48%. None of the objective-extreme configurations performs best across all criteria. The objective-extreme configurations reported in Table 8 represent optimization outcomes within the proof-of-concept search bounds and should not be interpreted as directly deployable operating settings. Industrial implementation would require expert verification of the corresponding parameter values and admissible bounds. The reevaluated Pareto-front approximation therefore provides candidate operating configurations for stakeholder-driven selection. Section 4.6 applies one illustrative post-search preference rule and evaluates the resulting scenario in the reference model.

4.6. NSGA-II Scenario Selection and Reference-Model Evaluation

Following NSGA-II optimization, a post-search selection rule was applied to the reevaluated Pareto-front approximation. Candidate configurations were required to retain at least 99% of the mean number of completed products obtained at the baseline configuration and not exceed the baseline-configuration values for grid-energy consumption or CO_2 emissions in the extracted CDT. Among the configurations satisfying these conditions, the configuration with the lowest equally weighted sum of normalized grid-energy consumption and CO_2 emissions was selected. Grid-energy consumption and CO_2 emissions were min–max normalized over the complete reevaluated Pareto-front approximation before their equally weighted sum was calculated.

In the extracted CDT, the selected scenario produced 70.87 completed products, consumed 65.56 kWh, and generated 2558.92 g of CO_2. Relative to the baseline configuration, completed products decreased by 0.27%, grid-energy consumption by 0.12%, and CO_2 emissions by 2.83%. The selected scenario, therefore, preserved production substantially better than the minimum-energy and minimum-emission configurations, which reduced completed products by more than 32%.

We then transferred the five idle-to-shutdown thresholds and two routing weights to the reference model while keeping all other parameters unchanged. We evaluated the baseline and selected scenarios using 100 independent replications of the 480-min production shift. Fig. 13 compares the resulting distributions, and Table 9 reports the means, 95% CIs, and relative changes. Mean completed products decreased from 68.61 to 68.41, corresponding to a reduction of 0.29%, while grid-energy consumption increased from 62.14 to 62.24 kWh, corresponding to an increase of 0.17%. The CIs for both differences included zero. The mean CO_2 emissions decreased from 2482.50 to 2387.96 g, corresponding to a reduction of 3.81%; the 95% CI for the difference, from −170.02 to −19.07 g, excluded zero. The mean throughput decreased from 0.940 to 0.930. The selected scenario received slightly more arriving orders on average, 73.71 compared with 73.03, while completed products remained nearly unchanged. The lower throughput may therefore partly reflect the higher number of arriving orders within the fixed simulation horizon.

Overall, the reference-model evaluation reproduced the qualitative pattern identified by the CDT, with completed products preserved, a minimal change in grid-energy consumption, and reduced CO_2 emissions. The selected routing policy assigned slightly more production to the lower-emission but higher-energy welding machine, which is consistent with the observed reduction in emissions without a corresponding reduction in grid-energy consumption. These findings support the proof-of-concept feasibility of the optimization-ready CDT for identifying candidate operating scenarios that can be evaluated in the reference model, but they do not demonstrate agreement across the decision space.

The selected scenario should be interpreted as the outcome of the illustrative post-search preference rule rather than as a generally preferred operating configuration. The 99% production-retention requirement defines the set of eligible configurations, while the equal weighting of normalized grid-energy consumption and CO_2 emissions determines the ranking within this set. Alternative production-retention thresholds or relative objective weights may lead to the selection of a different configuration from the same reevaluated Pareto-front approximation. Retaining the reevaluated Pareto-front approximation enables alternative stakeholder preferences to be applied without repeating the NSGA-II search, provided that the objective definitions and evaluated system conditions remain applicable.

4.7. Weighted-Sum Scalarization Optimization Results

To provide a comparative baseline for the Pareto-based optimization, the equal-weight scalarization defined in Section 4.4 was optimized using the GA. The scalarized search followed the settings summarized in Table 7. At termination, the best search-stage scalarized configuration was $x_\lambda^{search} = (5.6974, 10.9302, 0.4160, 17.8257, 0.2157, 0.9898, 0.0058)$. Based on the five search-stage replications, the configuration had a scalarized objective estimate of approximately $\widehat{J}(x_\lambda^{search}) = 0.1398$. The same decision vector was subsequently reevaluated using $r_{final} = 100$ independent replications. The baseline configuration was reevaluated using the same higher-replication seed set, and the objective components and scalarized objective were recomputed from the updated estimates. After higher-replication reevaluation, the scalarized configuration produced 48.53 completed products, a throughput of 0.649, grid-energy

Table 10
Baseline and weighted-sum scalarization results after 100-replication reevaluation.

Configuration	Completed products	Throughput	Grid-energy consumption (kWh)	CO_2 emissions (g)	$\widehat{J}$	Change in completed products	Change in energy	Change in emissions
Baseline configuration	71.06	0.935	65.64	2633.56	0.3333	–	–	–
Equal-weight scalarization	48.53	0.649	48.40	1159.93	0.1649	−31.71%	−26.27%	−55.96%

consumption of 48.40 kWh, and CO_2 emissions of 1159.93 g. The corresponding scalarized objective value was

$$\widehat{J}(x_{\lambda}^{*}) = 0.1649,$$

compared with

$$\widehat{J}(x_{base}) = \frac{1}{3} = 0.3333$$

for the baseline configuration. Relative to the 100-replication baseline, the scalarized configuration reduced grid-energy consumption by 26.27% and CO_2 emissions by 55.96%, while the number of completed products decreased by 31.71%. Table 10 summarizes the higher-replication results for the baseline and weighted-sum scalarized configurations.

The scalarized configuration was located close to the low-emission region identified by NSGA-II. In particular, the NSGA-II minimum-emission configuration produced 47.91 completed products, consumed 47.94 kWh, and generated 1139.60 g of CO_2, whereas the equal-weight scalarized configuration produced 48.53 completed products, consumed 48.40 kWh, and generated 1159.93 g of CO_2. The similarity indicates that the equal-weight scalarization identified a configuration in the low-emission region of the trade-off space, with substantial reductions in grid-energy consumption and CO_2 emissions accompanied by reduced production performance.

5. Discussion

The principal contribution of the proposed approach is the transformation of an automatically extracted, simulation-ready CDT into an optimization-ready CDT. From an industrial implementation perspective, the framework separates information that can be derived systematically from operational data and the extracted model from information requiring expert specification and approval. This separation enables candidate configurations to be evaluated before changes are implemented in the physical system while retaining expert control over technically admissible decisions. The case study demonstrates the feasibility of integrating the optimization-ready model with both a Pareto-based MOEA and weighted-sum scalarization. The scalarization experiment provides a comparative baseline under equivalent case-study settings; however, the study does not constitute a broad optimization-algorithm benchmark and does not report repeated independent optimization runs. Consequently, the results do not establish the general superiority or convergence reliability of either optimization approach.

The case-study results highlight several implications of this transformation. The results support the following observations:

- **Interpretation of the case-study findings.** The reduction from 80 search-stage nondominated configurations to 70 after higher-replication reevaluation illustrates the sensitivity of stochastic Pareto-front approximations to sampling variability. In the case study, the decision vectors remained unchanged, while their objective values were re-estimated using 100 rather than five independent replications before nondominance was recomputed. The change in nondominance status highlights the importance of higher-replication reevaluation before stakeholder preferences are applied.
- **Semi-automated preparation and expert involvement.** Parameters represented explicitly in MDPNML, such as routing weights and timing thresholds, can be identified and mapped systematically. However, model representation does not establish operational controllability. Experts must specify admissible bounds, physical dependencies, prohibited combinations, safety requirements, objectives, and operational limits before parameters can be modified. Future implementations could additionally record parameter provenance, units, controllability, confidence, update frequency, and expert approval.
- **Choice of optimization method and decision responsiveness.** The comparative scalarization baseline illustrates the implications of introducing objective preferences before rather than after optimization. Equal-weight scalarization returned one preference-dependent configuration, reducing grid-energy consumption and emissions by 26.27% and 55.96%, respectively, while reducing completed products by 31.71%. Weighted-sum scalarization requires the objective weights to be specified before optimization, and changing the weights generally requires a new optimization run [59, 62]. In contrast, the Pareto-based NSGA-II search retained multiple trade-off configurations, allowing stakeholder preferences to be introduced after higher-replication reevaluation [59,60]. When stakeholder priorities change while the objective definitions, system structure, decision space, and corresponding performance estimates remain applicable, another configuration can be selected from the existing reevaluated Pareto-front approximation without repeating the optimization search. This capability can reduce decision latency when priorities change frequently, or decisions are time-sensitive. Conversely, scalarization remains suitable when priorities are sufficiently stable, and a single preference-dependent solution is required. Surrogate-assisted optimization addresses a different concern by reducing the number of direct simulation evaluations when such evaluations are computationally expensive [67]. The appropriate optimization approach therefore depends on preference stability, decision urgency, simulation cost, system complexity, and the need to examine multiple trade-offs.
- **Context-aware scenario selection.** Industrial preferences may depend on external information such as electricity prices, grid-carbon intensity, renewable-energy forecasts, storage conditions, demand, machine availability, and planned or unexpected outages. Such information can alter preference thresholds, objective priorities, or operational constraints. Changes in contextual values that alter objective values require reevaluation of the stored configurations, whereas changes in system dynamics, constraints, or structure require a new optimization run.
- **Scalability to larger manufacturing systems.** The present case study considers seven continuous decision variables and three objectives, so scalability to larger optimization problems remains to be evaluated. For instance, the decision-support framework can be extended to mixed-variable decision spaces and additional constraints by expanding the optimization metadata and decision-to-model mapping to the corresponding controllable model parameters. For larger manufacturing systems, the increased simulation and optimization complexity may require computational enhancements

such as parallel simulation or surrogate-assisted optimization. Such extensions are particularly relevant to high-performance manufacturing systems, where multiple and potentially conflicting requirements for productivity, flexibility, resilience, reliability, and sustainability may need to be considered simultaneously [98]. The adaptability of the framework is also relevant to high-entropy manufacturing systems, which aim to manage individualized and dynamically changing demand through flexibility, reconfigurability, and adaptability [99].

- **Extension toward semi-physical commissioning.** A further deployment pathway is the integration of the optimization-ready CDT with Digital Twin-based remote semi-physical commissioning [100]. Candidate configurations identified through CDT-based optimization could first be evaluated in an environment combining virtual system components with selected physical equipment or controllers before implementation in the complete manufacturing system. During initial deployment, a human-in-the-loop stage could provide expert verification and approval of recommended changes. With sufficient operational validation, selected changes that remain within predefined technical and safety constraints could subsequently be automated. Such a staged approach could provide a transition from simulation-based decision support toward safe physical deployment.
- **Extension toward Industry 5.0 decision support.** The multidimensional modeling and optimization capabilities of the optimization-ready CDT could be extended to support broader Industry 5.0 priorities. For instance, at the modeling level, the MDSPN could represent human-involved manufacturing activities and human–machine interactions, with measurable human-centric aspects such as operator workload, ergonomic exposure, or intervention time incorporated as additional dimensions where appropriate data are available. At the optimization level, the optimization-preparation layer could incorporate sustainability, resilience, and human-centric performance requirements as objectives or constraints according to the decision context [101]. The executable multidimensional CDT and its optimization outputs could also provide a model-based component for future industrial-metaverse decision environments, where Digital Twins support virtual–physical integration and immersive interaction [102].

6. Summary and Outlook

In this paper, we introduced a framework for transforming an automatically extracted, simulation-ready Comprehensive Digital Twin into an optimization-ready Comprehensive Digital Twin for simulation-based multi-objective decision support. The framework applies Multi-Flow Process Mining (MFPM) to extract and parameterize dimension-specific stochastic Petri nets from multidimensional event logs and integrate them into a unified Multidimensional Stochastic Petri Net (MDSPN) represented in Multidimensional Petri Net Markup Language. A semi-automated optimization-preparation layer identifies model-represented parameters, incorporates expert-defined controllability information and admissible ranges, and maps decision vectors to executable model configurations. The framework supports both Pareto-based and scalarized optimization workflows; the case study instantiates the Pareto-based workflow using the Nondominated Sorting Genetic Algorithm II (NSGA-II) and evaluates weighted-sum scalarization using a Genetic Algorithm (GA) as a comparative baseline.

We demonstrated the complete workflow through an illustrative manufacturing case study, covering model extraction and validation, optimization preparation, Pareto-based search, higher-replication reevaluation, and scenario selection. The extracted model preserved the main production logic and showed agreement with the reference model for the selected KPIs at the baseline configuration. The NSGA-II optimization considered grid-energy consumption, CO_2 emissions, and the number of completed products without introducing objective weights or additional simulation-based performance constraints. After higher-replication reevaluation, 70 of the 80 search-stage nondominated configurations remained nondominated. The resulting Pareto-front approximation represented alternative trade-offs among the production and environmental objectives. We transferred one selected scenario to the reference model and evaluated it using 100 independent replications. Compared with the baseline scenario, the mean number of completed products decreased by 0.29%, from 68.61 to 68.41, while the mean grid-energy consumption increased by only 0.17%, from 62.14 to 62.24 kWh. Mean CO_2 emissions decreased by 3.81%, from 2482.50 to 2387.96 g. The reference-model evaluation therefore reproduced the intended qualitative outcome of the scenario selected using the Comprehensive Digital Twin, with completed products preserved, a negligible energy effect, and reduced emissions. An additional equal-weight scalarization baseline reduced grid-energy consumption and CO_2 emissions by 26.27% and 55.96%, respectively, while reducing completed products by 31.71% relative to the extracted-model baseline. This result illustrates the effect of specifying objective preferences before optimization. These findings support the proof-of-concept feasibility of the proposed framework as a systematic, executable, and interpretable basis for simulation-based multi-objective decision support.

Future research will focus on four directions: (i) evaluating the framework using real-world manufacturing data, live event streams, and industrial production systems, including human-in-the-loop verification and approval of recommended configuration changes; (ii) investigating larger mixed-variable decision spaces and additional objectives and constraints, (iii) developing context-aware decision support by incorporating, for instance, time-dependent electricity prices, grid-carbon intensity, production demand, machine availability, storage conditions, planned and unexpected outages, and weather and renewable-generation forecasts; and (iv) extending the MDSPN and optimization-preparation framework toward Industry 5.0 by representing human-involved manufacturing activities and measurable human-centric and resilience-related performance requirements as additional modeled dimensions, objectives, or constraints.

CRediT authorship contribution statement

Atieh Khodadadi: Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Formal analysis, Data curation, Conceptualization. **Sanja Lazarova-Molnar:** Supervision. **Xueping Li:** Supervision.

Declaration of Competing Interest

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

Acknowledgments

The authors extend their thanks for the funding received from the ONE4ALL project funded by the European Commission, Horizon Europe Programme under Grant Agreement No. 101091877.

Code Availability

The source code, Multidimensional Petri Net Markup Language models, optimization metadata, and scripts used for the case study are available in the accompanying repository [97].

References

[1] Lasi H, Fettke P, Kemper H-G, Feld T, Hoffmann M. "Industry 4.0. Bus & Inf Syst Eng 2014;6(4):239–42.
[2] Abikoye OC, et al. "Application of internet of thing and cyber physical system in Industry 4.0 smart manufacturing. Emergence of Cyber Physical System and IoT in Smart Automation and Robotics: Computer Engineering in Automation. Springer; 2021. p. 203–17.
[3] Helu M, Libes D, Lubell J, Lyons K, Morris KC. Enabling smart manufacturing technologies for decision-making support. In: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. American Society of Mechanical Engineers; 2016. p. V01BT02A035.
[4] Mihai S, et al. "Digital twins: A survey on enabling technologies, challenges, trends and future prospects. IEEE Commun Surv & Tutor 2022;24(4):2255–91.
[5] Kritzinger W, Karner M, Traar G, Henjes J, Sihn W. "Digital Twin in manufacturing: A categorical literature review and classification. Ifac-Pap 2018; 51(11):1016–22.
[6] Negri E, Fumagalli L, Macchi M. "A review of the roles of digital twin in CPS-based production systems. Procedia Manuf 2017;11:939–48.
[7] Lazarova-Molnar S. A vision for advancing digital twins intelligence: key insights and lessons from decades of research and experience with simulation. Proc 14th Intern Conf on Simulation and Modeling Methodologies, Technologies Applications. Dijon, France: SCITEPRESS-Science and Technology Publications; 2024. p. 5–10.
[8] Kasilingam S, Yang R, Singh SK, Farahani MA, Rai R, Wuest T. "Physics-based and data-driven hybrid modeling in manufacturing: a review. Prod & Manuf Res 2024;12(1):2305358.
[9] Van der Auweraer H, Hartmann D. arXiv preprint. *arXiv:2210.17402,* Exec Digit Twin Merging Digit Phys Worlds 2022.
[10] Khodadadi A. "Essential Data Requirements for Industrial Energy Efficiency with Digital Twins: A Case Study Analysis. 7th Int Conf Emerg Data Ind Hasselt Belg 2024.
[11] Balbo G. "Introduction to stochastic Petri nets. School Organized by the European Educational Forum. Springer; 2000. p. 84–155.
[12] Lugaresi G, Zanotti M, Tarasconi D, Matta A. "Manufacturing systems mining: Generation of real-time discrete event simulation models. 2019 IEEE Int Conf On Systems, Man Cybernetics (SMC). IEEE; 2019. p. 415–20.
[13] Khodadadi A, Zare A, Michelle J, Götz M, Lazarova-Molnar S. A tutorial on data-driven petri net model extraction and simulation for Digital Twins in smart manufacturing. Presente Winter Simul Conf (WSC) Seattle WA USA 2025.
[14] Breque M, Nul LDe, Petridis A. "Industry 5.0: towards a sustainable, human-centric and resilient European industry. Dir General Res Innov (DG RTD) Eur 2021.
[15] G. Erbach, "European climate law," in "Regulation (EU)," European Parliamentary Research Service (EPRS), European Parliament, PE 649.385, 2021, vol. 1119. [Online]. Available: https://www.univiu.org/images/aauniviu2017/GP/co-curr/European_climate_law.pdf.
[16] EU. "Regulation (EU) 2021/1119 of the European Parliament and of the Council of 30 June 2021 establishing the framework for achieving climate neutrality and amending Regulations (EC) No 401/2009 and (EU) 2018/1999 ('European Climate Law'). J Eur Union L 243 2021;1:64.
[17] J. Cresko et al., "US department of energy's industrial decarbonization roadmap," U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy (EERE), DOE/EE-2635, 2022. [Online]. Available: https://www.osti.gov/servlets/purl/1961393.
[18] Wolter K, Zisowsky A. On Markov reward modelling with FSPNs. Perform Eval 2001;44(1-4):165–86.
[19] Wang Q, Wang X, Yang S. "Energy modeling and simulation of flexible manufacturing systems based on colored timed petri nets. J Ind Ecol 2014;18(4): 558–66.
[20] Khodadadi A, Lazarova-Molnar S. "Multi-flow Process Mining for Comprehensive Simulation Model Discovery. Presente ICICM 2024.
[21] Khodadadi A, Lazarova-Molnar S. "Multi-Flow Process Mining as an Enabler for Comprehensive Digital Twins of Manufacturing Systems. Winter Simulation Conference, 2025. Seattle, Washington: IEEE; 2025.
[22] Khodadadi A, Lazarova-Molnar S. "Automated extraction of comprehensive digital twin models for smart manufacturing systems. J Manuf Syst 2026;85: 287–306.
[23] Khodadadi A, Lazarova-Molnar S. "MDPNML: A Multidimensional Petri Net Markup Language Enabling Construction and Simulation of Comprehensive Digital Twin Models. Int Conf Model-Based Softw Syst Eng (MODELSWARD) 2026.
[24] Zhou A, Qu B-Y, Li H, Zhao S-Z, Suganthan PN, Zhang Q. "Multiobjective evolutionary algorithms: A survey of the state of the art. Swarm Evolut Comput 2011;1(1):32–49.
[25] Miettinen K. Nonlinear multiobjective optimization. Springer Science & Business Media; 1999.
[26] Jahn J. "Scalarization in multi objective optimization. Mathematics of Multi Objective Optimization. Springer; 1985. p. 45–88.
[27] Cozad A, Sahinidis NV, Miller DC. "Learning surrogate models for simulation-based optimization. AIChE J 2014;60(6):2211–27.
[28] Ali M, Alarjani A, Mumtaz M. "A NSGA-II based approach for multi-objective optimization of a reconfigurable manufacturing transfer line supported by Digital Twin: A case study. Adv Prod Eng Manag 2023;18:116–29.
[29] Zhu Q, Huang S, Wang G, Moghaddam SK, Lu Y, Yan Y. "Dynamic reconfiguration optimization of intelligent manufacturing system with human-robot collaboration based on digital twin. J Manuf Syst 2022;65:330–8.
[30] Guo H, Chen M, Mohamed K, Qu T, Wang S, Li J. "A digital twin-based flexible cellular manufacturing for optimization of air conditioner line. J Manuf Syst 2021;58:65–78.
[31] Deb K, Pratap A, Agarwal S, Meyarivan T. "A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evolut Comput 2002;6(2):182–97.
[32] Gray J, Rumpe B. "Models in simulation. Softw & Syst Model 2016;15(3):605–7.
[33] Varga A. "Discrete event simulation system. Proc Eur Simul Multiconference (ESM'2001) 2001;17.
[34] Naugle A, Langarudi S, Clancy T. "What is (quantitative) system dynamics modeling? Defining characteristics and the opportunities they create. Syst Dyn Rev 2024;40(2):e1762.
[35] Helbing D. "Agent-based modeling. Agent-based Simulations and Experiments Study Emergent Social Behavior. Social self-organization. Springer; 2012. p. 25–70.
[36] Brailsford SC, Eldabi T, Kunc M, Mustafee N, Osorio AF. "Hybrid simulation modelling in operational research: A state-of-the-art review. Eur J Oper Res 2019; 278(3):721–37.
[37] Wooley A, Bitencourt J, Silva D. Bridging the gap between discrete event simulation and digital twin: A manufacturing case study. Manuf Lett 2025;44: 1274–84.
[38] Grieves M. "Completing the cycle: Using PLM information in the sales and service functions. SME Management Forum, 19. NJ, USA: SME Forum South Orange; 2002. p. 3.
[39] Yao J-F, Yang Y, Wang X-C, Zhang X-P. "Systematic review of digital twin technology and applications. Vis Comput Ind Biomed Art 2023;6(1):10.
[40] Friederich J, Francis DP, Lazarova-Molnar S, Mohamed N. "A framework for data-driven digital twins of smart manufacturing systems,". Comput Ind 2022;136: 103586.
[41] Côté P-O, Nikanjam A, Ahmed N, Humeniuk D, Khomh F. "Data cleaning and machine learning: a systematic literature review. Autom Softw Eng 2024;31(2): 54.
[42] Fanaee-T H, Gama J. "Event labeling combining ensemble detectors and background knowledge. Progress Artif Intell 2014;2(2):113–27.
[43] Van Der Aalst W. "Process mining. Commun ACM 2012;55(8):76–83.
[44] Friederich J, Khodadadi A, Lazarova-Molnar S. "PySPN: A Python Library for Modeling, Simulation, and Event Log Generation of Stochastic Petri Nets," *Transactions of The Society for Modeling and.* Simul Int 2025;2025.
[45] Van Der Aalst W. "Data science in action. Process Mining: Data Science in Action. Springer; 2016. p. 3–23.
[46] Bano D, Michael J, Rumpe B, Varga S, Weske M. "Process-aware digital twin cockpit synthesis from event logs. J Comput Lang 2022;70:101121.
[47] Pourbafrani M, van der Aalst WM. "Data-Driven Simulation In Process Mining: Introducing A Reference Model. ECMS; 2023. p. 411–20.
[48] Castiglione C. "Automated generation of digital models for manufacturing systems: The event-centric process mining approach. Comput & Ind Eng 2024; 197:110596.
[49] Kumbhar M, Ng AH, Bandaru S. "A digital twin based framework for detection, diagnosis, and improvement of throughput bottlenecks. J Manuf Syst 2023;66: 92–106.
[50] Murata T. "Petri nets: Properties, analysis and applications. Proc IEEE 2002;77(4): 541–80.
[51] Mejía G, Pereira J. "Multiobjective scheduling algorithm for flexible manufacturing systems with Petri nets. J Manuf Syst 2020;54:272–84.
[52] He X, Murata T. "High-level Petri nets—extensions, analysis, and applications. The Electrical Engineering Handbook. Elsevier; 2005. p. 459–75.
[53] Bonnefoi F, Choppy C, Kordon F. "A discretization method from coloured to symmetric nets: application to an industrial example. Transactions on Petri Nets and Other Models of Concurrency III. Springer; 2009. p. 159–88.
[54] F. Bause and P.S. Kritzinger, Stochastic petri nets. Vieweg Wiesbaden, 2002.
[55] Rogge-Solti A, van der Aalst WM, Weske M. "Discovering stochastic petri nets with arbitrary delay distributions from event logs. International Conference on Business Process Management. Springer; 2013. p. 15–27.
[56] S. Lazarova-Molnar, "The proxel-based method: Formalisation, analysis and applications," PhD dissertation, Otto-von-Guericke-Universität Magdeburg, 2005.
[57] Amaran S, Sahinidis NV, Sharda B, Bury SJ. "Simulation optimization: a review of algorithms and applications. Ann Oper Res 2016;240(1):351–80.
[58] Hunter SR, et al. "An introduction to multiobjective simulation optimization. ACM Trans Model Comput Simul (TOMACS) 2019;29(1):1–36.
[59] Marler RT, Arora JS. "Survey of multi-objective optimization methods for engineering. Struct Multidiscip Optim 2004;26(6):369–95.
[60] Emmerich MT, Deutz AH. "A tutorial on multiobjective optimization: fundamentals and evolutionary methods. Nat Comput 2018;17(3):585–609.
[61] Das I, Dennis JE. "A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. Struct Optim 1997;14(1):63–9.
[62] Marler RT, Arora JS. "The weighted sum method for multi-objective optimization: new insights. Struct Multidiscip Optim 2010;41(6):853–62.
[63] Kim IY, De Weck OL. "Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. Struct Multidiscip Optim 2005;29(2): 149–58.
[64] Jiang P, Zhou Q, Shao X. "Surrogate-model-based design and optimization. Surrogate Model-based Engineering Design and Optimization. Springer; 2019. p. 135–236.

[65] Jin Y. "Surrogate-assisted evolutionary computation: Recent advances and future challenges. Swarm Evolut Comput 2011;1(2):61–70.
[66] Díaz-Manríquez A, Toscano G, Barron-Zambrano JH, Tello-Leal E. "A review of surrogate assisted multiobjective evolutionary algorithms,". Comput Intell Neurosci 2016;2016(1):9420460.
[67] Knowles J. "ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. IEEE Trans Evolut Comput 2006;10(1):50–66.
[68] Deb K. "Multi-objective evolutionary algorithms. Springer Handbook of Computational Intelligence. Springer; 2015. p. 995–1015.
[69] Laarhoven PJ, Aarts EH. Simulated annealing: theory and applications. Springer; 1987.
[70] Glover F. "Tabu search: A tutorial. Interfaces 1990;20(4):74–94.
[71] Deb K. "Multi-objective optimisation using evolutionary algorithms: an introduction. Multi-objective Evolutionary Optimisation for Product Design and Manufacturing. Springer; 2011. p. 3–34.
[72] E. Zitzler, Evolutionary algorithms for multiobjective optimization: Methods and applications. Shaker Ithaca, 1999.
[73] Zitzler E, Laumanns M, Thiele L. "SPEA2: Improving the strength Pareto evolutionary algorithm. TIK Report 2001;103.
[74] Beume N, Naujoks B, Emmerich M. "SMS-EMOA: Multiobjective selection based on dominated hypervolume. Eur J Oper Res 2007;181(3):1653–69.
[75] Zhang Q, Li H. "MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evolut Comput 2007;11(6):712–31.
[76] Deb K, Jain H. "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. IEEE Trans Evolut Comput 2013;18(4):577–601.
[77] Wang X, Zhang F, Yao M. "A many-objective evolutionary algorithm with metric-based reference vector adjustment. Complex & Intell Syst 2024;10(1):207–31.
[78] Ishibuchi H, Imada R, Setoguchi Y, Nojima Y. "Performance comparison of NSGA-II and NSGA-III on various many-objective test problems. 2016 IEEE Congress on Evolutionary Computation (CEC). IEEE; 2016. p. 3045–52.
[79] Kalyanmoy D. "A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans Evolut Comput 2002;6(2):182–97.
[80] Wu C, Cai J, Liu X, Shi H, Fu J. "Green dynamic scheduling optimization and visual monitoring system for micro-hole drilling production line based on digital twin. Int J Comput Integr Manuf 2025:1–25.
[81] Dai M, Shen Y, Liu W, Lü M. "Digital twin-based optimization of AGV scheduling under energy constraints in plant factories,". J Clean Prod 2025;519:146016.
[82] Schmitt T, Juan SO, Amouzgar K, Hanson L, Moris MU. "Optimizing energy efficiency and productivity in industrial manufacturing: A simulation-based optimization approach with knowledge discovery. J Manuf Syst 2025;82:748–65.
[83] Zan X, Wu Z, Guo C, Yu Z. "A Pareto-based genetic algorithm for multi-objective scheduling of automated manufacturing systems. Adv Mech Eng 2020;12(1):1687814019885294.
[84] Amodeo L, Chen H, Hadji AEl. "Multi-objective supply chain optimization: An industrial case study. Workshops on Applications of Evolutionary Computation. Springer; 2007. p. 732–41.
[85] Sharda B, Banerjee A. "Robust manufacturing system design using multi objective genetic algorithms, Petri nets and Bayesian uncertainty representation. J Manuf Syst 2013;32(2):315–24.
[86] Marsan MA. "Stochastic Petri nets: an elementary introduction. European Workshop on Applications and Theory in Petri Nets. Springer; 1988. p. 1–29.
[87] Sanders WH, Meyer JF. "A unified approach for specifying measures of performance, dependability and performability. Dependable Computing for Critical Applications. Springer; 1991. p. 215–37.
[88] Jensen K, Kristensen LM, Wells L. "Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. Int J Softw Tools Technol Transf 2007;9(3):213–54.
[89] Schuster D, van Zelst SJ, van der Aalst WM. "Utilizing domain knowledge in data-driven process discovery: A literature review. Comput Ind 2022;137:103612.
[90] Khodadadi A, Lazarova-Molnar S. "Multidimensional Stochastic Petri Nets: A Novel Approach to Modeling and Simulation of Stochastic Discrete-Event Systems (Vienna, Austria) Presente IEEE Int Conf Syst Man Cybern (SMC) 2025 (Vienna, Austria).
[91] Billington J, et al. "The petri net markup language: Concepts, technology, and tools. International Conference on Application and Theory of Petri Nets. Springer; 2003. p. 483–505.
[92] Sargent RG. "Verification and validation of simulation models. Proceedings of the 2010 Winter Simulation Conference. IEEE; 2010. p. 166–83.
[93] Goldsman D, Nelson BL, Banks J. "Comparing systems via simulation. Handb Simul Princ Methodol Adv Appl Pract 1998:273–306.
[94] A. Borshchev, "The big book of simulation modeling: multimethod modeling with AnyLogic 6," (No Title), 2013.
[95] Holland JH. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. The MIT press; 1992.
[96] Blank J, Deb K. "Pymoo: Multi-objective optimization in python. IEEE Access 2020;8:89497–509.
[97] A. Khodadadi and S. Lazarova-Molnar. Optimization-Ready Comprehensive Digital Twin. (2026). v1.0.0. [Online]. Available: https://github.com/atikh/Optimization-Ready-Comprehensive-Digital-Twin. doi: 10.5281/zenodo.22015005.
[98] Leng J, et al. "High-performance manufacturing systems: concepts, performance metrics, enablers, challenges, and research directions. Adv Eng Inform 2025;68:103617.
[99] Leng J, et al. "A sustainable mass individualization paradigm of high-entropy manufacturing systems towards Industry 5.0: retrospect and prospect. J Manuf Syst 2026;88:519–43.
[100] Leng J, et al. "Digital twins-based remote semi-physical commissioning of flow-type smart manufacturing systems. J Clean Prod 2021;306:127278.
[101] Lu Y, et al. "Outlook on human-centric manufacturing towards Industry 5.0. J Manuf Syst 2022;62:612–27.
[102] Guo J, et al. "Industrial metaverse towards Industry 5.0: Connotation, architecture, enablers, and challenges. J Manuf Syst 2024;76:25–42.