



**HAL**  
open science

# Leveraging digital twin and dynamic scheduling for enhanced human–robot collaboration

Pierre Hémono, Ahmed Nait Chabane, M’hammed Sahnoun

## ► To cite this version:

Pierre Hémono, Ahmed Nait Chabane, M’hammed Sahnoun. Leveraging digital twin and dynamic scheduling for enhanced human–robot collaboration. *International Journal of Production Economics*, 2025, pp.109823. <10.1016/j.ijpe.2025.109823>. <hal-05316670>

**HAL Id: hal-05316670**

**<https://hal.science/hal-05316670v1>**

Submitted on 15 Oct 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

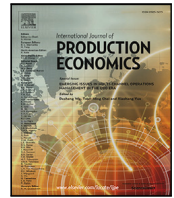


Distributed under a Creative Commons CC BY 4.0 - Attribution - International License



Contents lists available at ScienceDirect

## International Journal of Production Economics

journal homepage: [www.elsevier.com/locate/ijpe](http://www.elsevier.com/locate/ijpe)Leveraging digital twin and dynamic scheduling for enhanced human–robot collaboration<sup>☆</sup>Pierre Hémono<sup>a,b,d</sup><sup>\*</sup>, Ahmed Nait Chabane<sup>b</sup>, M'hammed Sahnoun<sup>c</sup><sup>a</sup> eXcent, 101 rue henri gautier, Montoir de Bretagne, 44550, Pays de la Loire, France<sup>b</sup> CESI, Le Paquebot, 24, Saint-Nazaire, 44600, Pays de la Loire, France<sup>c</sup> CESI, 80 avenue Edmund Halley, Rouen Madrillet Innovation, Saint-Étienne-du-Rouvray, 76800, Normandie, France<sup>d</sup> Department of Engineering Sciences, University of Agder, Jon Lilletuns vei 9, Grimstad, 4879, Agder, Norway

## ARTICLE INFO

Dataset link: <https://github.com/PierreHemono/Dynamic-Behavior-Tree/tree/main>

## Keywords:

Human–robot collaboration  
Digital twin  
Industry 5.0  
Human factors

## ABSTRACT

Industry 5.0 represents a paradigm shift toward human-centric, resilient, and sustainable production systems. At the core of this transformation lies digital twins, which enable predictive and prescriptive analytics in real time, improving decision-making capabilities such as visibility, transparency, and collaboration. By integrating advanced AI algorithms for data interpretation and facilitating seamless human–machine interactions, digital twins address critical challenges in modern industrial systems. This article explores the transformative role of digital twins in operational decision-making, focusing on their ability to optimize workflows, and foster collaboration between humans and robots. Through a dual-layer methodology macro-level task scheduling for efficiency and consideration of human factors and micro-level real-time control for adaptability, digital twins offer a powerful framework for aligning human and robotic capabilities while mitigating human fatigue and improving decision transparency. Highlighting applications in digital transformation, optimization, and human–AI collaboration, this study emphasizes how digital twins enhance operational visibility and resilience. The findings contribute to the evolution of Industry 5.0, offering innovative solutions for integrating predictive models and human-centered approaches in decision-making, redefining the future of sustainable and collaborative industrial systems.

## 1. Introduction

The Fifth Industrial Revolution introduces a range of new prospects and difficulties, underscoring the necessity of positioning humans as the central focus in industrial operations. To accomplish this, it is necessary for manufacturers to aim for improved adaptability of existing workshop equipment, thus reinforcing humans essential contribution.

Human–robot collaboration (HRC) is a key tool for accomplishing this transition toward an increased human presence in the production process (George and George, 2020). However, effectively enabling this collaboration requires meticulous planning and execution of the collaborative work design.

Creating collaborative applications in robotics encounters naturally intricate obstacles. Primarily, collaborative robots are employed to automate activities, which frequently results in a reduced involvement of human workers. These difficulties arise from various factors, including how well robots are integrated and accepted by human workers in manufacturing environments, adherence to implementation standards,

and ergonomic aspects of both software and hardware elements (Rega et al., 2021). Designing applications for collaborative robotics involves inherent complexities, necessitating meticulous planning and execution. Furthermore, research indicates that human considerations are frequently neglected in the design of human–robot interactions (Hémono et al., 2025; Sgarbossa et al., 2020), resulting in challenges when re-integrating humans into the process. To address these challenges, it has become essential to develop innovative approaches for task allocation that optimize seamless integration of cobots in workshops. This involves not only improving productivity but also improving the working conditions of human operators. By assigning high-value tasks to humans, collaboration between human workers and cobots can lead to increased worker satisfaction and the cultivation of valuable skills. Trust is also a critical factor in the collaboration between humans and robots, as highlighted in Chen et al. (2020), and is essential for user acceptance of robotic systems. The reliability and predictability of a coworker, whether human or robotic, are paramount in high-value

<sup>☆</sup> This article is part of a Special issue entitled: 'Digital Twins in Operations and Supply Chains' published in International Journal of Production Economics.

<sup>\*</sup> Corresponding author at: Department of Engineering Sciences, University of Agder, Jon Lilletuns vei 9, Grimstad, 4879, Agder, Norway.

E-mail address: [pierre.a.hemono@uia.no](mailto:pierre.a.hemono@uia.no) (P. Hémono).

<https://doi.org/10.1016/j.ijpe.2025.109823>

Received 19 December 2024; Received in revised form 22 August 2025; Accepted 8 October 2025

Available online 11 October 2025

0925-5273/© 2025 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

workstations where errors can incur significant costs for the employing companies. Ensuring effective collaboration necessitates that both human and robotic collaborators meet the same stringent standards of reliability and predictability to maintain seamless and productive operations.

In recent decades, advances in perception, reasoning, and artificial intelligence (AI), particularly machine learning and deep learning, have significantly enhanced robot capabilities (Premebida et al., 2018). AI plays an essential role in Industry 5.0, facilitating the development of smarter and more efficient manufacturing processes (Zafar et al., 2024). These innovations enable robots to plan and execute complex actions, adapt to dynamic environments, and collaborate effectively within diverse human–robot teams, playing essential roles in trajectory planning and intelligent control.

Despite these advancements, trust in robotic systems continues to be limited, as the actions carried out by robots are often not clear or anticipated by human coworkers (De Graaf et al., 2021). The need for flexibility and transparency grows with the increase in responsibilities and tasks assigned to robots. The seamless human–robot collaboration requires the equipping of robots with cognitive capabilities that enable their awareness of the environment (Dimitropoulos et al., 2021). The significance of cognitive systems in improving the efficiency of automated operations has been recognized, particularly in the context of assembly and disassembly tasks, as highlighted by Li et al. (2018), Choux et al. (2021). This is increasingly pertinent as the demand for customized products grows and the necessity for material reuse transforms production material supply dynamics. A production line must be versatile enough to handle a myriad of products to maintain cost-effectiveness. Although there have been advancements, there is still ample potential for enhancement, thus making it an ongoing research challenge.

To improve the clarity and comprehension of robot actions and consequently strengthen the trust of human partners in human–robot interaction, actions must be organized in a logical sequence. A predictable and systematic flow of actions is critical for building trust among human collaborators working with robots. Effectively addressing these requirements requires planning on both macro and micro levels. Lou et al. (2024) shows that very few studies simultaneously consider task scheduling (including allocation and sequencing) and task execution. Among those who do, none address both the specific abilities of humans and robots and the unique requirements of collaborative joint tasks.

This framework can be seen as a dual-level configuration addressing the identified challenges: establishing a robust human–robot collaboration scheme and ensuring adaptability and responsiveness to occurring events.

At the macro level, scheduling models such as the Flexible Job Shop Scheduling (FJSP) and the Assembly Line Balancing Problem (ALBP) play pivotal roles in coordinating the collaborative efforts of humans and robots. These models ensure that tasks are allocated in a way that optimizes overall workflow and productivity while considering human comfort and ergonomics (Hémono et al., 2023).

The importance of ergonomics in activity scheduling is well established. Well-designed ergonomic considerations can yield a return on investment between 1:2 and 1:10 within 24 months. Furthermore, corrective actions due to ergonomic errors in workplace design are significantly more expensive, being 9.2 times more expensive than preventive actions, as highlighted in Ghorbani et al. (2023), Hendrick (2008), and Falck and Rosenqvist (2014). Studies such as (Kheirabadi et al., 2023) have illustrated that, despite an increased focus on human factors (HF) such as ergonomics and safety over the last 10 years, around 25% of the research on task scheduling in ALBP from 2017 to 2023 and about 37% of the articles on task scheduling in Collaborative Robotic Assembly Line Balancing Problems (CRALBP) from 2019 to 2023 still do not address dynamic control solutions crucial for

Industry 5.0. This deficiency in dynamic control affects the responsiveness and adaptability of robots, which is essential for building human trust. Robots need to demonstrate intelligent behavior and real-time adaptability to operator needs in order to become more reliable and predictable, thereby fostering human trust.

On a micro level, the control framework for executing robot actions is essential for planning robot activities in accordance with operational requirements. Among various control frameworks, BTs have demonstrated superior performance and responsiveness in dynamic environments. Due to their modular structure, BTs have surpassed Hierarchical Finite State Machines (HFSMs) in terms of performance, readability, and ease of design, as shown in Iovino et al. (2022). They are also easier to extend, reuse, and adapt. The automatic generation of BTs for robotic action planning facilitates the execution of detailed and adaptable task schemes. BTs dynamically adapt to robot tasks and environmental interactions, improving their responsiveness and reliability in their roles. This highlights the necessity of advanced and adaptive action planning methods as a foundation for achieving resilient human–robot interaction in dynamic industrial settings. Several methods exist to automatically generate BTs, including learning methods, evolutionary methods, and high-level planning methods such as Planning Domain Definition Language (PDDL), Task and Motion Planning (TAMP), or Hierarchical Tasks Networks (HTN). Each method presents distinct benefits for structuring and organizing robotic tasks, resulting in more intuitive and reliable human–robot interactions. By offering greater flexibility and responsiveness, BTs address trust concerns, ensuring consistent and comprehensible behaviors. The 20 challenges highlighted in Karupiah et al. (2023) emphasize the necessity for a combined approach of high-level (macro) and low-level (micro) planning to enhance human–robot collaboration (HRC) in Industry 5.0. These challenges underscore the importance of improving flexibility, reactivity, and trust in robotic systems to achieve effective and safe human–robot interactions.

The automation of behavior design and testing concerning the reliability of decision-making in robots operating within collaborative frameworks can be conceptualized as a Digital Twin (DT). This approach highlights the potential for resilience and flexibility in dynamic scheduling and control, which are essential elements that are often lacking in the successful integration of collaborative robots in industrial settings (Zafar et al., 2024).

In summary, there is a critical need for advanced planning and control methods that address the limitations of traditional scheduling approaches while enhancing the flexibility, reactivity, and safety of human–robot collaborations.

DT is experiencing significant growth with a market predicted to reach 35 billion by 2025 (Marketsandmarkets.com, 2020). DT has first been used to test on a virtual prototype without affecting real operations, for example, in maintenance process as in Roy et al. (2016). DT considering human factors has started to spread since 2015, but only from 2019 there is a notable acceleration in considering this matter (Berti and Finco, 2022). This acceleration is related to the emerging paradigm of industry 5.0 which, since 2020, shows interest in human-centricity at the heart of activities (E. Commission, D.-G. for Research, Innovation et al., 2021). Digital Twin, as found in Berti and Finco (2022), allows the conceptualization of systems where the data are shared both from the physical system to the numeric model and conversely. The use of DT combined to a scheduling model with consideration of human factors represents an opportunity to improve the dynamic control at micro-level and ensure that the realization of optimal plan is possible even if non-predictable events arrive in the environment.

This paper addresses the integration of macro-level scheduling using high-level optimization with micro-level execution modeled through a Digital Twin incorporating automatically generated Behavior Trees. The focus is on exploring how this dual-level approach can enhance

human–robot collaboration by improving adaptability, transparency, and trust in robotic systems.

Given these challenges and the current state of research, this study seeks to address the following research questions:

1. How can digital twins (DTs) and dynamic scheduling enhance human–robot collaboration in industrial settings ?
2. What are the gaps in the current literature on digital twins, particularly concerning human factors and dynamic control ?
3. How can behavior trees (BTs) be utilized to improve the adaptability and transparency of robotic systems in collaborative environments ?

To frame this discussion, a concise state-of-the-art review will be presented that identifies two key areas. Digital Twins addressing human factors in manufacturing and the automatic generation of Behavior Trees. DTs are examined for their potential to bridge the gap between current practices and the requirements of advanced human–robot collaboration, emphasizing human-centric design and interaction.

The paper will also explore Behavior Trees as a tool for dynamic robotic behavior programming. Their ability to manage goals, sub-goals, and preconditions in real time makes them a valuable component for improving responsiveness in task execution. By integrating automatically generated BTs into a Digital Twin framework, the proposed approach creates a dynamic and adaptive system. This integration not only enables robots to respond effectively to unforeseen environmental changes but also provides a clear and interpretable structure for task execution, fostering operator trust and collaboration in shared environments.

The main contributions of this work are as follows:

- **Integration of High-Level Scheduling with a Digital Twin:** Development of a pipeline combining high-level optimization ([Gurobi Optimization, LLC \(2023\)](#)) with a dynamic digital twin represented as an automatically generated Behavior Tree (BT), enhancing real-time visibility and collaboration.
- **Improved Transparency and Human Trust:** The dynamic generation of BTs from scheduling plans ensures explainable robot behavior, fostering human–robot interaction and trust in collaborative environments.
- **Memory-Driven Robotic Systems:** Introduction of a memory mechanism that allows the robot to retain a history of successes and failures, improving decision-making and adaptability to achieve goals.

The rest of the article is structured as follows. Section 3 details the Methodology, introducing the dual-level framework and the integration of macro-level scheduling with micro-level execution using a Digital Twin and Behavior Trees. Section 4 presents the experimental validation of the framework, including the results and their implications. Section 5 highlights managerial insights, emphasizing the practical benefits and opportunities offered by this approach to improve human–robot collaboration. Finally, Section 6 concludes with a summary of the contributions and describes perspectives for future research.

## 2. Literature review

Building on the challenge sketched out in Section 1 that Industry 5.0 must simultaneously drive resilient throughput and safeguard worker well-being under constant shop-floor disruption, we organize existing work into three interconnected streams. We begin by examining Digital Twins, the bidirectional virtual–physical mirror that can ingest human-factor signals (ergonomics, safety, cognitive load) and translate them into actionable insights. Next, we turn to Dynamic Scheduling, the macro-level engines that leverage both machine telemetry and human-centered metrics to replan operations in real time. Finally, we review Behavior Trees, the on-board, reactive control structures robots use

to carry out, or locally replan, scheduled tasks when disturbances strike. Although each pillar boasts a rich standalone literature, to our knowledge, no existing framework binds them all human-aware DTs, real-time schedulers, and auto-generated BT replanners into a unified, two-tier architecture. That integration is precisely the gap our work tries to fill.

### 2.1. The digital twin in the era of industry 5.0

According to [Grieves \(2014\)](#), there are three foundational elements that currently define a digital twin (DT), a virtual model, its physical counterpart, and the bidirectional link between them ([Grieves, 2014](#); [Sharma et al., 2022](#)). Subsequent surveys ([Kritzinger et al., 2018](#); [Berti and Finco, 2022](#)) categorize DTs by their levels of integration:

- **The Digital Model:** In this setup, data transfers are managed manually from the virtual model to its physical counterpart and back.
- **The Digital Shadow:** Here, the data flow is manually initiated from the virtual model to the physical model but automatically from the physical model back to the digital model.
- **The Digital Twin:** This represents the highest level of integration, where data transmission is fully automated in both directions.

Despite extensions of the basic DT concept e.g. service/data taxonomies ([Tao et al., 2018a](#)) or formal V&V (Verification and validation) workflows ([Tao et al., 2018b](#)) detailed, large scale case studies remain rare. As [Sharma et al. \(2022\)](#) and [Kritzinger et al. \(2018\)](#) observe, practical implementations are often described only at a high level, with few end to end deployments providing the depth needed for reproducibility or cross-industry transfer.

Nonetheless, when fully implemented, DTs can generate substantial economic value. For instance, Thomas ([Thomas, 2024](#)) estimates that widespread DT adoption in U.S. manufacturing could yield median annual cost savings of \$27.2 billion (90% CI: \$16.1–38.6 billion) through improved asset utilization, predictive maintenance, and enhanced decision support. Similarly, [Bányai and Kovács \(Bányai and Kovács, 2023\)](#) report that an RFID-enabled DT system achieved a 9.2% higher ROI and 53% greater NPV compared to a conventional barcode-based setup. [Malik et al. \(2024\)](#) demonstrate up to a 35% increase in operational efficiency via better scheduling, shorter cycle times, and faster shop-floor reaction. These concrete findings directly motivate our techno-economic framework (Section 4.1), where we apply multi-year annuitization and NPV/ROI analysis to quantify the business case for mobile manipulators.

Although digital-twin research has surged since 2019, a systematic review of 35 human-factors-DT studies found that only about 20% address true human–robot collaboration ([Berti and Finco, 2022](#); [Sgarbossa et al., 2020](#)). The bulk of work falls into three clusters:

- **Ergonomic risk assessment** - posture and reach analysis with RULA/REBA indices or exoskeleton support models ([Arisoy et al., 2016](#); [Constantinescu et al., 2019](#); [Ippolito et al., 2020](#));
- **Worker safety** - in shared workcells, including dynamic collision warnings and safe-zone supervision ([Bilberg and Malik, 2019](#); [Havard et al., 2019](#));
- **Mental/physical workload** - monitoring stress, fatigue, and cognitive load via biosensors, cameras, and AI inference ([Bécue et al., 2020](#); [Tarallo et al., 2018](#); [Pusch and Noël, 2019](#)). Only a handful of frameworks such as [Peruzzini et al.'s](#) bidirectional ergonomic-DT ([Peruzzini et al., 2020](#)), [Greco et al.'s](#) real-time posture monitoring ([Greco et al., 2020](#)), and [Ogunseju et al.'s](#) exposure management twin [Ogunseju et al. \(2021\)](#) actually weave these human factors into live DT control for dynamic task reallocation and on-the-fly re-planning.

While [Peruzzini et al. \(2020\)](#) demonstrate a bidirectional DT that maps worker postures to virtual avatars and back to adjust ergonomic exposure, their framework stops short of using those posture updates to drive on-the-fly task reallocation. [Greco et al. \(2020\)](#) deliver real-time posture monitoring and alert generation within a DT, improving operator comfort, yet they do not integrate these alerts into a live scheduler or re-planning loop. [Ogunseju et al. \(2021\)](#) present a self-management DT that tracks cumulative ergonomic load and suggests recovery breaks, but it operates in isolation from production planning and cannot automatically redistribute tasks when limits are reached.

In all three cases, human-factor data enrich the digital twin, but none embed those metrics into a unified, online scheduling engine nor support automated, real-time re-planning under shop-floor disturbances.

As such, while a handful of works embed posture, workload or safety metrics into digital-twin prototypes, to our knowledge no existing framework simultaneously integrates those human-factor measurements into a live macro-level scheduler and triggers automatic, real-time micro-level re-planning for example via generated Behavior Trees to preserve both throughput and operator well-being under shop-floor disturbances.

## 2.2. The dynamic scheduling in industry 5.0

While independent digital twins (DTs) monitor the real-time conditions of both humans and machinery, dynamic scheduling systems integrate these data streams to reallocate tasks instantaneously. In contrast to static Gantt charts, a genuinely dynamic scheduler adapts in real-time by incorporating unexpected events (such as equipment failures, employee absences, or urgent orders) and human-related factors (like ergonomic strain and fatigue levels). This allows for constant adjustments, aligning with the resilience required by Industry 5.0 every few seconds or minutes ([Ouelhadj and Petrovic, 2009](#); [Destouet et al., 2023](#); [Renke et al., 2021](#)).

Dynamic scheduling methods are typically grouped into three main families ([Ouelhadj and Petrovic, 2009](#)):

- **Reactive methods** deploy dispatching rules or learning-based agents on the fly, without a predefined full schedule ([Sahnoun et al., 2019](#)). Common priority rules (e.g. Earliest Due Date, Shortest Process Time, Critical Ratio) exhibit widely varying resilience across disruption types.
- **Pro-active methods** embed robustness at inception via safety buffers or robust optimization formulations to mitigate the need for frequent re-planning ([Nait Chabane et al., 2023](#)).
- **Predictive-reactive hybrids** combine prognostics (machine learning or simulation) with localized re-planning triggers when forecasted risk thresholds are exceeded ([Bouaziz et al., 2024](#)).

[Renke et al. \(2021\)](#) demonstrate the evolution of techniques from basic dispatching rules to advanced heuristics and metaheuristics, with a continuous incorporation of predictive and robustness capabilities. The majority of studies in the literature focus on static scheduling issues because dynamic scheduling is significantly more complex to manage. Among the studies that tackle dynamic scheduling, most concentrate on macro-level scheduling challenges, with or without considering human-factor metrics ([Sgarbossa et al., 2020](#); [Hémono et al., 2025](#)), or emphasize solely robot-level controllers such as Behavior Trees, Model Predictive Control (MPC), or reactive policies without integrating feedback from the scheduler.

To tackle the uncertainty present in production systems, a promising approach involves implementing a hybrid of proactive-reactive methods, incorporating a dual-layer strategy consisting of offline scheduling (proactive) and online scheduling (reactive) ([Wu et al., 1999](#); [Sanogo et al., 2025](#)). Researchers assert that systems which integrate knowledge-based scheduling with interactive scheduling methods to

produce a viable plan are inherently complex, challenging to sustain and modify, inflexible, expensive, and tardy in meeting personalized customer demands ([Lou et al., 2012](#)).

To fully realize Industry 5.0, which emphasizes throughput, resilience, and the well-being of workers, it is imperative to marry strategic, human-focused planning with real-time, robot-centered re-planning. To our knowledge, our two-tier framework is the first to incorporate operator well-being into both the macro scheduler and the deployment of an auto-generated, on-board Behavior Tree for local re-planning, thereby ensuring feedback integration between strategic planning and dynamic execution amid uncertainty.

## 2.3. The automatic generation of behavior tree

Behavior Trees serve as decision-making frameworks in robotic systems, game engines, or cognitive architectures, directing agent behaviors according to a specific state ([Francis, 2017](#); [Iovino et al., 2022](#)). BTs are comprised of execution nodes, which are also called leaves, and control nodes, known as non-leaves, with their root typically depicted on the left side or at the top of the diagram.

BTs are extensively employed to manage NPC (non-playable character) behaviors in video games. They arrange tasks in a hierarchical manner, partitioning actions into simple (atomic) or complex (composite) forms. This structure underlies their widespread use in both gaming and robotics due to their effective and adaptable nature ([Francis, 2017](#)). Compared to Finite-State Machines (FSMs), BTs boast superior responsiveness and adaptability. Incorporation of reactive nodes, known as Event-Driven Behavior Trees (EDBT), improves interaction by allowing systems to react to events in real time, thus enhancing coordination between agents ([Agis et al., 2020](#)). A notable benefit of BTs is their modular nature, which facilitates a flexible and expandable design for sophisticated robotic initiatives, in contrast to the more inflexible FSMs ([Pi et al., 2023](#)). Additionally, studies have shown that real-time decision-making in unpredictable settings can be optimized ([Abiyev et al., 2016](#)) through the use of supplementary techniques like fuzzy logic.

In addition, the use of BTs as a control framework helps reduce uncertainty in robotic systems. In [Nkisi-Orji et al. \(2023\)](#), a method is proposed to address uncertainties encountered by BT-driven agents by reusing previously unsuccessful solutions to generate new ones. This approach, known as failure-driven transformational case reuse, improves problem-solving efficiency and reduces errors related to uncertainty.

The automated generation of BTs constitutes a pivotal advancement in the field of intelligent autonomous systems. Numerous strategies have been formulated to endow autonomous agents with adaptable and efficient behaviors, thereby addressing the lack of formal design methodologies noted in [Colledanchise et al. \(2017\)](#). These strategies encompass planning methods that exploit pre-existing behaviors to augment flexibility and reusability, as well as machine learning algorithms that deduce behaviors through environmental interactions. Each approach embodies unique advantages and limitations, and hybrid techniques have emerged to amalgamate these methods and mitigate individual shortcomings.

Although BTs were initially developed to manage video game characters, their potential in robotics remains underexplored. This section provides a comprehensive overview of methods for generating BTs across various domains.

The goal of this study is to explore how automatic BTs generation can enhance human-robot collaboration in different contexts. Thanks to their structured and modular nature, BTs have become a popular choice in research. A literature review reveals three main methods for automatically generating BTs, with the following approaches being the most frequently employed:

- The learning methods.
- The genetic programming method.

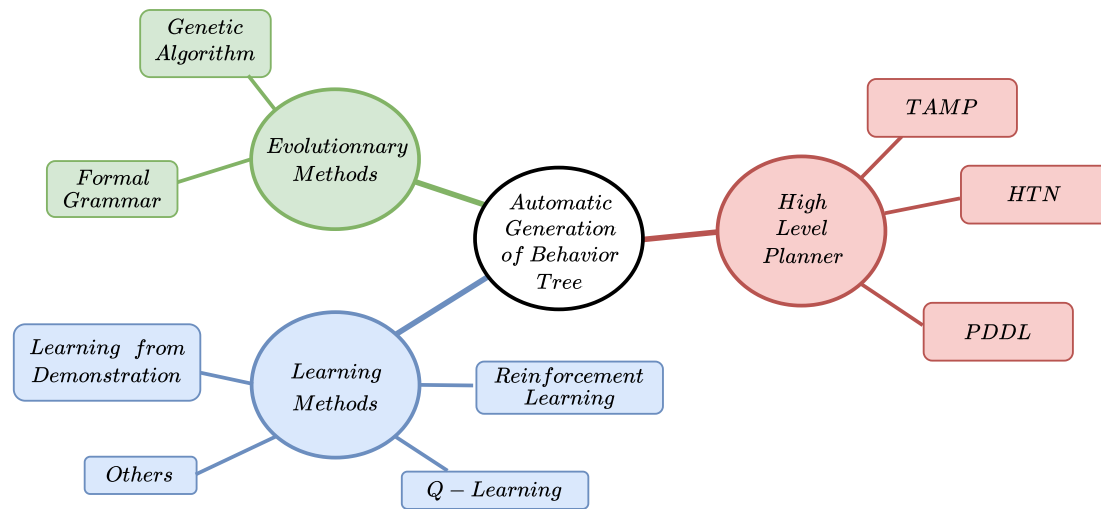


Fig. 1. Main methods for Automatic Generation of Behavior Tree.

- The method using high-level planners.
- The remaining are automatically generated using heuristics or different frameworks.

Learning-based methods excel in dynamic or poorly defined settings, where structuring workflows via Behavior Trees (BTs) proves invaluable. Two dominant paradigms have emerged for automatic BT synthesis (Fig. 1):

**Learning from Demonstration (LfD)** - clones expert behavior by observing and imitating human operators. French et al. (2019) show that LfD can yield highly modular, reactive BTs, and Scherf et al. (2023) demonstrate that interactive refinement at runtime further enhances flexibility. However, LfD's success hinges on demonstration quality, and it does not inherently solve macro-level questions of *who* does *what* or *when*.

**Reinforcement Learning (RL)** - often via Q-learning lets agents discover optimal action sequences through trial and error. Dworzan-ski and Hlavacs (2024) automatically evolve BT structures, Child and Dey (2013) embed Q-conditions to sharpen decision-making, and Banerjee (2018) produce interpretable BTs with minimal manual tuning. R. d. P. Pereira and Engel (2015) integrate RL nodes for real-time adaptation. Yet RL introduces its own challenges, complex implementation, reward-shaping burdens, and overfitting risks that can hinder robustness.

Both LfD and RL significantly cut down manual BT design and boost reactivity, but neither explicitly aligns micro-level BT adaptation with macro-level task scheduling or human-factor metrics. This gap suggests a promising opportunity: to develop hybrid frameworks that combine *macro* planners (e.g. PDDL-based schedulers) with *micro* BT generators enriched by LfD/RL and to feed back ergonomic and workload measurements so that re-planning dynamically preserves both performance and operator well-being.

Evolutionary algorithms, notably Genetic Programming (GP) and Grammatical Evolution (GE) treat BT generation as a search over populations of candidate trees, refined via mutation, crossover, and selection. GP directly evolves tree structures (Iovino and Smith (2023), Klöckner (2013)), while GE imposes a grammar to constrain and guide the evolution. Their modular, hierarchical nature makes BTs especially suitable to these operations.

Empirical studies illustrate their effectiveness:

- Lim et al. (2010) show GP-evolved BTs outperform hand-crafted ones in the DEFCON simulation (albeit with overfitting risks),

- Colledanchise et al. (2018) and Perez et al. (2011) validate GP/GE BTs on the Mario AI benchmark, demonstrating superior adaptability over finite-state models,
- Scheper et al. (2016) and Iovino et al. (2021) extend these methods to real-world tasks — drone navigation and mobile manipulation — highlighting robustness in unpredictable environments.

However, evolutionary methods demand careful fitness-function design, incur high computational cost for large populations (Tao et al. (2020)), and may produce BTs that are hard to interpret (Estgren and Jansson (2017)).

In the context of Industry 5.0, integrating evolutionary methods into a broader task planning framework could provide significant benefits. For example, on an assembly line with changing product configurations, evolutionary BTs generation could enable robots to autonomously adjust strategies, reducing downtime and increasing productivity. To fully realize these advantages, it is essential to manage computational resources, carefully design fitness functions, and avoid overfitting. By overcoming these challenges, evolutionary methods can produce robust modular BTs that enhance human-robot collaboration in industrial settings.

Among approaches for automatic Behavior Tree (BT) generation, a significant strategy involves the use of high-level planners such as the Planning Domain Definition Language (PDDL), found in more than half of the reviewed literature. PDDL provides a structured framework to define the robot's knowledge domain, skills, and action parameters, making it easier to create BTs that meet the desired objectives. These planners produce symbolic plans that can be automatically transformed into BTs, combining deliberative task allocation with reactive control.

For example, Iovino and Smith (2023) leverage PDDL to define robot capabilities and derive BTs via backward chaining. Martín et al. (2021) use PlanSys2 to convert PDDL plans into BT, enabling efficient execution of parallel actions and dynamic re-planning under ROS2. This ensures enhanced flexibility and reactivity, which is essential for adapting to changing environments and human-robot interactions.

Task and motion planning (TAMP) solutions can integrate abstract task definitions with kinematic solvers, as shown by Ruiz-Celada et al. (2022), improving adaptability in dynamic scenarios by coordinating high-level tasks and low-level motions.

Beyond PDDL-based methods, other formalisms have also proven to be effective. Colledanchise et al. (2017) synthesize BTs from Linear Temporal Logic (LTL) specifications for formal correctness. Converting

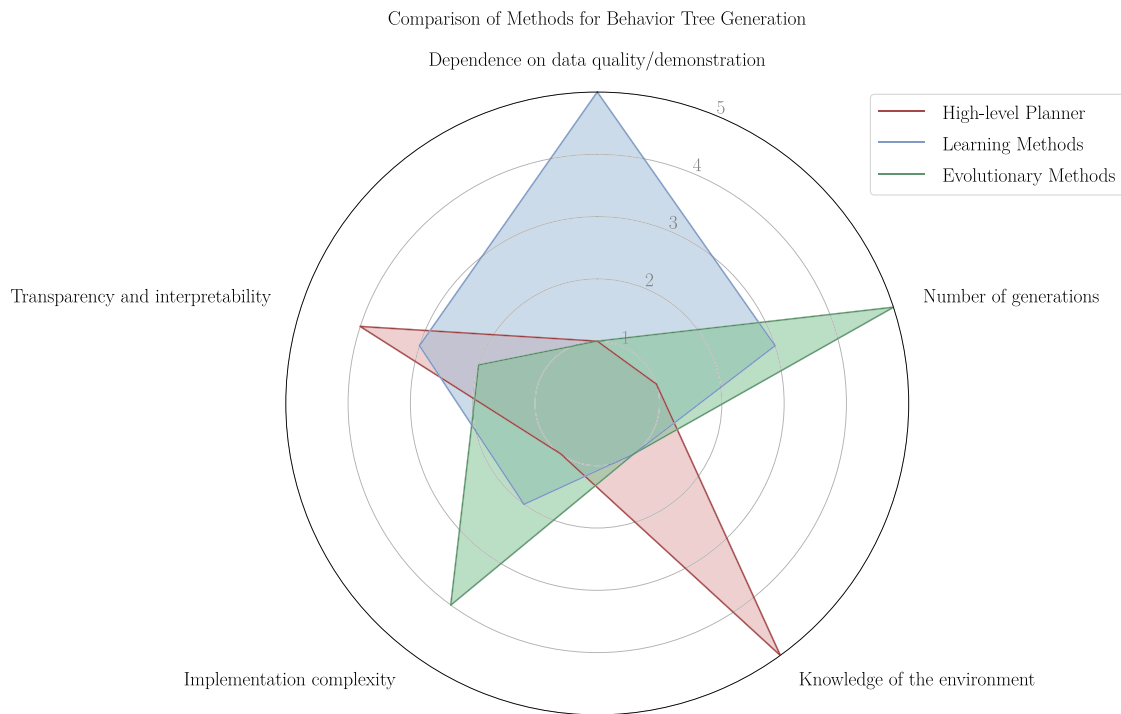


Fig. 2. Strength and weakness of each methods for automatically generating BTs.

hierarchical task networks (HTN) into BTs, as demonstrated by Segura-Muros and Fernández-Olivares (2017), can enhance modularity and reduce manual design.

Overall, these high-level planning approaches increase reactivity and flexibility, allowing agents to adapt to complex tasks and dynamic environments. Nevertheless, their performance depends on accurate domain definitions and the planner's configuration. Planning can be computationally demanding, and the complexity of generated BTs may raise trust and safety concerns in industrial contexts.

In conclusion, high-level planners provide a promising avenue for automatic BTs generation, reducing manual effort while increasing adaptability. To succeed in Industry 5.0 environments, careful resource management and reliability assurance are essential, fostering operator confidence and safe, efficient human-robot collaboration.

The synthesis of the main features of each method is illustrated in Fig. 2. Notations range from 0, indicating no concern or impact from the feature, to 5, signifying a high level of concern or impact.

#### 2.4. Synthesis and discussion of state of the art

Despite the rapid proliferation of digital-twin (DT) research, there remains no single, universally adopted definition, and current implementations continue to exhibit characteristics of Industry 4.0, emphasizing performance indicators and equipment efficiency rather than comprehensive integration of human factors. Some efforts concentrate on spot-checking ergonomic risks or generating “shadows” of worker postures, but they stop short of closing the loop into live decision support or respecting ethical concerns such as voluntary consent and privacy. As a result, DTs too often function as passive observers rather than active partners in a truly human-centric, Industry 5.0 ecosystem.

Complementing DTs, dynamic scheduling promises the real-time resilience that static Gantt charts cannot provide. By continuously ingesting live data streams, machine breakdowns, urgent orders or human metrics like fatigue and ergonomic strain, reactive dispatch rules, proactive buffers and hybrid predictive-reactive frameworks can each adapt on-the-fly. Yet in isolation they falter: purely reactive schemes may overload operators, robust-buffer methods waste capacity, and hybrids introduce complexity that slows replanning. Crucially,

none natively marries ergonomic or human-wellness criteria to rapid, moment-to-moment replanning.

Moving forward, the integration of human-centered scheduling with automated BTs generation techniques, such as those based on PDDL, offers promising avenues for improving human-robot collaboration. Rather than monitoring and modeling the human operator in a way that could raise ethical concerns or introduce psychological pressure, focusing on robot adaptability, performance, and resilience can indirectly serve human well-being and efficiency. By incorporating human factors into the initial design phase, as demonstrated in our previous work (Hémono et al., 2025), it becomes possible to create operational decision-making tools that ensure transparency, facilitate cooperation, and enhance worker comfort without continuous scrutiny. Combining human-aware scheduling with an automatically generated BTs policy not only supports flexible, dynamic task allocation, but also ensures that the resulting system aligns with Industry 5.0 principles. In this way, advanced predictive and prescriptive models can offer real-time adjustments, fostering trust, safety, and long-term sustainability in collaborative industrial environments.

In light of the challenges identified in Section 2.2 notably that purely reactive, proactive or hybrid scheduling methods tend to be overly complex, inflexible and slow to adapt (Lou et al., 2012) we see a clear opportunity to simplify the overall architecture. First, leveraging Digital Twins with integrated production planning, control, and layout capabilities, as suggested by Rosen et al. (2015) and Uhlemann et al. (2017), can shift the paradigm toward truly human-centric applications.

By incorporating mechanisms for detailed diagnostics and automatic planning triggered by both environmental conditions and operator decisions, Digital Twins can transcend their current efficiency-driven paradigms to promote human trust, adaptability, and ultimately well-being.

Secondly, the incorporation of these technological advancements with frameworks that prioritize user ergonomics, safety, and comfort establishes a foundation for tackling ethical and psychological issues. By utilizing a dual-layer scheduling approach, where the off-line, human-centered macro scheduling determines “what” and “when”

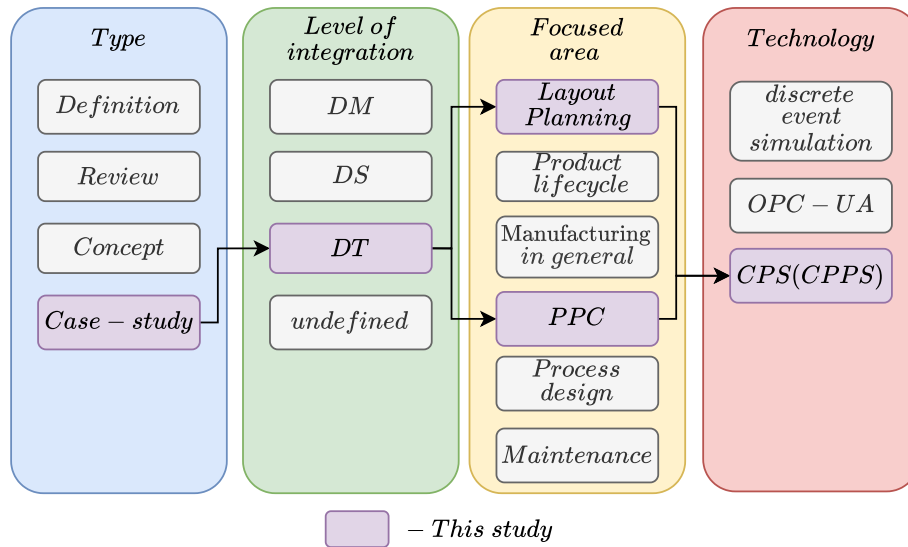


Fig. 3. Figure from (Kritzinger et al., 2018) with highlight on our study.

actions occur, paired with the online, BT-inspired micro replanning specifying "how" execution unfolds, the identified complexity gap can be effectively addressed (Wu et al., 1999; Aloulou and Portmann, 2005). This method should not only improve production efficiencies but also foster a sustainable, human-focused industrial environment, aligning with the evolution towards Industry 5.0 principles.

In the following sections, we will demonstrate how our study employs these opportunities, applying integrated Digital Twin concepts to enhance both production planning and layout control. In doing so, our goal is to improve decision support, foster trust, and create a more resilient, human-focused industrial ecosystem. Fig. 3, adapted from (Kritzinger et al., 2018), illustrates our position and the approach we propose in this study.

### 3. Methodology

The results of our previous research highlight the interest of a two-step planning approach integrating scheduling based on human parameters, as well as a robotic control structure derived from this planning. In this section, we will present the software structure enabling the conversion of the task execution plan into a control structure for robots, the demonstrator that allowed us to validate our hypotheses, as well as our testing protocol and the results obtained. We have chosen to use behavior trees that have proven their superior performance in robot control. The results of the state-of-the-art in Section 2 emphasizes the benefits of using planners to automatically generate BTs. These planners are easy to implement and their hierarchical structure enables organized action plans, mirroring how humans organize actions.

#### 3.1. Pipeline definition

We present a pipeline that translates task scheduling logs between human and robot through Gurobi (Gurobi Optimization, LLC, 2023) into PDDL, demonstrating the effectiveness of our two-level planning strategy. A solver processes the PDDL plan, converting it into a dynamic behavior tree for robot control, as detailed in Fig. 4.

The optimization model used in the pipeline is the one from a previous work (Hémono et al., 2025), where we defined a way to organize human-robot collaboration regarding two optimization objectives, the working difficulty level (WDL) and the makespan. In this work, we adopt the taxonomy of Özgüven et al. (2010), where *Job*, *Task* and *Operation* mirror the classical FJSP hierarchy, and we further define *Action* as the atomic Behavior Tree leaf (e.g. MoveTo, Grasp) that directly invokes each operation in the robot middleware.

For each specific case submitted to the Gurobi optimizer, it is necessary to provide the difficulty levels  $W_{ijk}$  and the nominal operating durations  $t_{ijk}$  as input data. According to Hémono et al. (2025), the difficulty levels are defined with respect to Karhu et al. (1977) and AFNOR (2005). The optimization logs are then analyzed and used to extract, via a converter:

- The task ID
- The operation ID
- The resource ID executing the operation (Human, Robot, Human and Robot)
- The task start time
- The task end time

The optimization performed through the Gurobi solver results in the generation of a log rich in optimization information. This file records all possible combinations of variables and parameters. Our main objective is to extract the information required to form the optimal path identified in the Gantt chart, i.e., the variables assigning a "True" or non-zero value. We also aim to eliminate duplicates resulting from the assignment of certain tasks to the dummy resource "Co".

The algorithm that outlines the pseudocode of our converter is available on GitHub. In Table 1, we demonstrate an example of data extracted and processed using the converter (P.1), as illustrated in the section "High-level planner - model optimization" of Fig. 4.

A dispatcher (referred to as P.2 in Fig. 4) integrates the readable logs obtained through the converter (referred to as P.1 in Fig. 4) with a manually generated database that associates jobs and operations with their constituent elementary actions. This integration results in a database consisting of two files: the domain, which contains predicates and actions to be executed along with their preconditions and effects, and the problem, which comprises objects such as tools, agents, stations, their initial situations, and the goal(s) to be achieved. This database is formalized into two files that feed the PDDL knowledge base: domain.pddl and problem.pddl. We assume that each elementary action is provided with a set of default preconditions and that the role of this dispatcher is to automatically direct input data to the appropriate file structure to automate the development of the robot action plan. It should be noted that the elementary actions that make up our tasks are relatively limited in number.

- Move (*move to* in the pseudo-code)
- Pick (*Pick* in the pseudo-code)
- Place (*Place* in the pseudo-code)

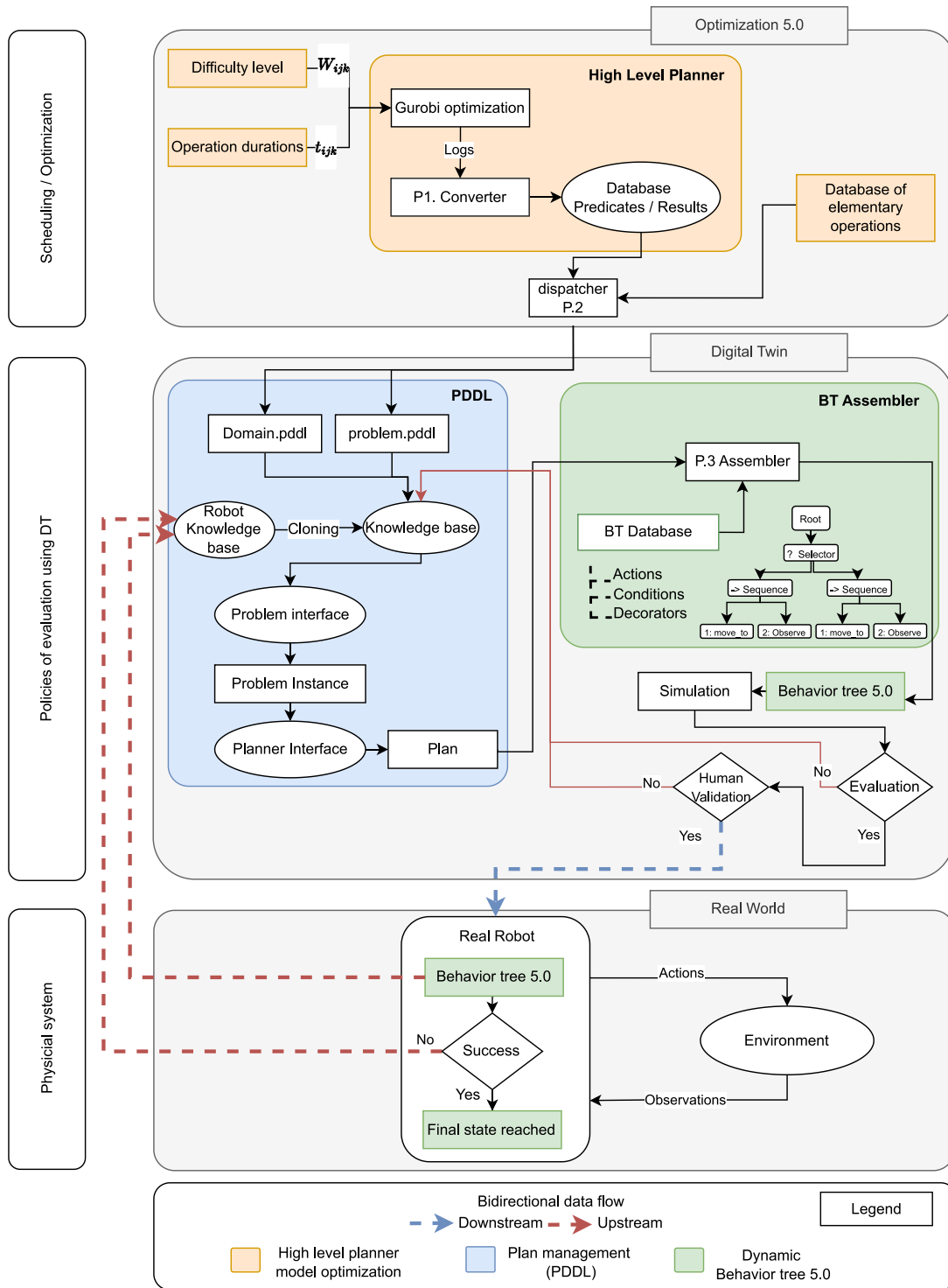


Fig. 4. Pipeline integrating a high-level planner with the digital twin.

- Observe (*look at* in the pseudo-code)

The collection of basic actions enables us to divide the tasks in our schedule into distinct actions. For instance, within an industrial context, the action of acquiring a tool can be described as follows:

- Move to the storage coordinates of the tools
- Observe the presence of the tools
- Pick up the tools
- Move to the assembly station

- Observe that the area is clear
- Place the tools

By reducing the number of successive elementary actions, tasks can be decomposed more efficiently. Limiting the number of elementary actions allows for the manual definition of generic preconditions without requiring excessive programming effort.

The pseudocode of algorithm P.2 Dispatcher is available on [GitHub](#). The automated creation of PDDL files, namely domain.pddl and problem.pddl, must result in files strict enough to prevent any deviation

**Table 1**  
Data extracted and reprocessed by the converter.

Operation	Job	Resource	Starting time (Sijk)	End time (Cijk)
op <sub>5</sub>	j <sub>1</sub>	R	0	208
op <sub>6</sub>	j <sub>2</sub>	H	20	32
op <sub>2</sub>	j <sub>1</sub>	H	35	50
op <sub>1</sub>	j <sub>1</sub>	H	54	203
op <sub>3</sub>	j <sub>1</sub>	R	208	591
op <sub>4</sub>	j <sub>2</sub>	H	451	509
op <sub>4</sub>	j <sub>1</sub>	H	591	828
op <sub>4</sub>	j <sub>1</sub>	R	591	915
op <sub>4</sub>	j <sub>1</sub>	Co	591	828
op <sub>7</sub>	j <sub>2</sub>	H	828	1206
op <sub>1</sub>	j <sub>2</sub>	R	967	1180
op <sub>2</sub>	j <sub>2</sub>	H	1206	1494
op <sub>3</sub>	j <sub>2</sub>	H	1494	1692
op <sub>5</sub>	j <sub>2</sub>	H	1692	2412
op <sub>1</sub>	j <sub>3</sub>	H	2412	2649
op <sub>1</sub>	j <sub>3</sub>	R	2412	2663
op <sub>1</sub>	j <sub>3</sub>	Co	2412	2659
op <sub>2</sub>	j <sub>3</sub>	H	2649	2669
op <sub>3</sub>	j <sub>3</sub>	H	2669	2759
op <sub>5</sub>	j <sub>3</sub>	R	2671	2903
op <sub>4</sub>	j <sub>3</sub>	H	2759	2904

from the initial plan before its first execution. The resulting plan is required to align closely with the schedule produced by the Gurobi solver, as it constitutes the optimal solution that minimizes both the makespan and the difficulty level for the human operator. However, these two files must also provide sufficient flexibility to allow for plan recalculation in the event of unexpected circumstances during task execution or environmental changes, even if this requires compromising the solution's quality. This is essential to develop an executable plan and prevent the robot from being stuck in a state of uncertainty.

Using a solver with PDDL in addition to the Gurobi solver can, if not well-suited to the problem type or specific needs, result in an inconsistent plan. We opt for the POPF3 solver (Partial-Order Planning Forward) (Coles et al., 2010), which is compatible with the PDDL version we are using. This solver aligns well with the strategy we have adopted, as it is specifically designed to handle problems involving multiple agents and diverse features. POPF3 employs a method known as forward chaining, meaning it starts from the initial state and works towards establishing a plan that reaches the final state by executing possible actions.

After its creation, the domain.pddl file remains unchanged, while the problem.pddl file is updated based on changes in the knowledge base. The adopted solution to meet the aforementioned constraints consists of several components.

- The chaining of effects and preconditions
- The definition of predicates specific to each operation in the plan
- The addition of generic actions

The chain of effects and preconditions ensures that the execution of the plan follows the order specified by the Gurobi optimization. This approach is illustrated in Fig. 5. One of the effects of the performed action is the completion of that specific action, and to ensure that an action N-1 precedes an action N, we simply insert this completion effect into the preconditions of the subsequent action. This method allows for maintaining the correct order of operations during the plan resolution. This approach is fully aligned with the nature of the solver used.

The use of this strategy requires the declaration of the effects of each specific operation in the domain.pddl predicates, as illustrated in Fig. 6.

To ensure the necessary flexibility for transitioning between the various states of the robot and for developing a new effective plan to overcome a blocking situation, we propose the introduction of a generic action. This action serves as a bridge between the degraded plan and

the initial plan for the robot's actions. For example, in the case where a tool is not at its expected location, the plan must include additional movements. Although the generic action is not an integral part of the sequence of operations leading to the final objective, it allows for the insertion of operations to restore the expected progression in the sequence. The introduction of this generic action, which differs from specific actions in its formulation and remains standard in the use of PDDL, is illustrated in Fig. 7.

The domain.pddl and problem.pddl files constitute the knowledge base of PDDL. The "problem interface" of PDDL generates an instance of the problem which is then solved to produce a plan. The resulting plan contains a list of actions sorted by execution order with specific information related to the operation to be performed, such as the tool used in the operation and the location of execution. An example of a plan is illustrated in Fig. 8. The domain.pddl and problem.pddl files are provided in initial PDDL.

The different steps of this plan are then constructed into a Behavior Tree using the Assembler (identified as P.3) in the BT Assembler block of Fig. 4. The algorithm is designed to automatically generate a BT from a formalized plan in PDDL. The algorithm consists of two steps: analyzing and extracting actions from a plan file, and then constructing the corresponding behavior tree.

The strategy is to encapsulate each pick, place, or move sequence within an "UpdateKB" decorator, which uses ROS messages to adjust the knowledge base (problem.pddl) based on "success" or "failure" states in the behavior tree. For instance, a "success" for a "move to" action triggers the addition of the new location and the removal of the previous one in the knowledge base. In this way, we can develop logical event management as well as, most importantly, a memory for the robot that executes the assigned tasks.

The assembler requires as input the plan generated in the previous step by Rosplan, as well as various manually entered and project-specific information, such as tool and equipment marker IDs and location coordinates.

Algorithm P.3 Assembler, begins by initializing a knowledge base containing the operations and marker identifiers, as well as a structure to organize the parameters of each predicate and a mapping of opposite locations. The function parsePddlPlan then extracts and organizes the actions from the PDDL plan by identifying their types (movements, picking, placing, etc.) and filtering out invalid or redundant actions through filterActions, thus optimizing the relevant set of actions.

From the filtered actions, the algorithm generates a behavior tree by identifying unique operations and predicates to create task sequences and associated decorators. The tree is structured hierarchically, with a root sequence grouping the generated behaviors for each action (movements, object handling, object placement). The tree generation concludes by writing the sequences and decorators to a Python file.

The main program combines these different functions to generate the final tree from the actions in the PDDL plan. Each elementary action is associated with a condition or decorator in the action base of the behavior tree. Depending on the nature of the operation, selector nodes are added to generate efficient and safe behavior, facilitating human-robot collaboration and reducing the need for re-planning. The decorators check the BT state ("success" or "failure") to propose updates to the database. Once deployed on a physical robot, the compiled BT runs onboard and directly drives the robot's interaction with the environment. The BT issues actions (e.g., move, pick, place) that affect the environment, while sensor feedback provides observations on the execution state. This perception-action loop ensures that the robot's behavior continuously reflects the real-world situation and connects the virtual planning layer with the physical execution layer.

The continuous update of the database after each action allows the robot to adjust its behavior in response to environmental changes, making the behavior tree more flexible and adaptive. In case of uncertainty or blockage, indicated by a "failure" state in the tree, a new plan is developed. For instance, if the robot does not find a tool at the expected

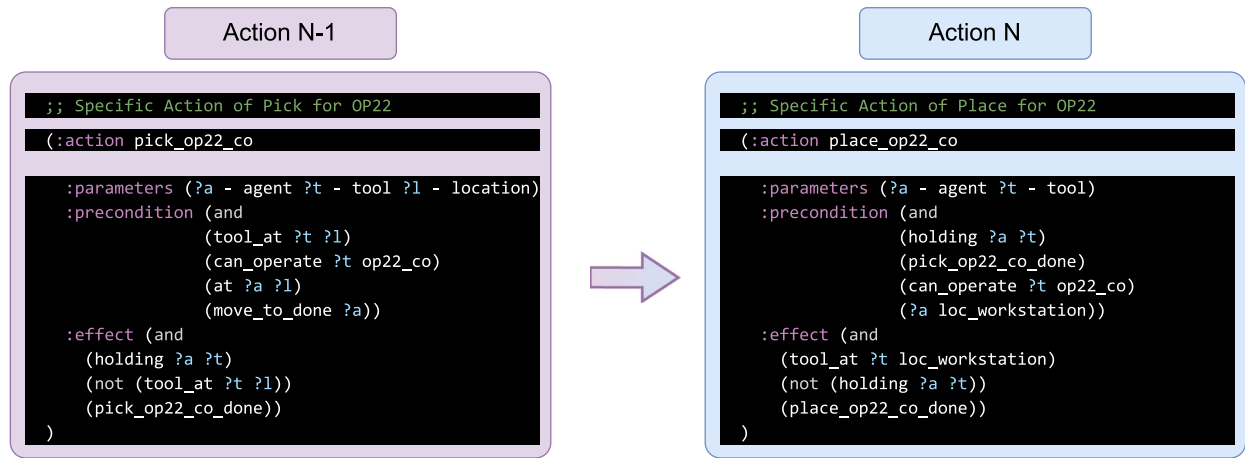


Fig. 5. Excerpt from domain.pddl illustrating the chaining of effects and preconditions.

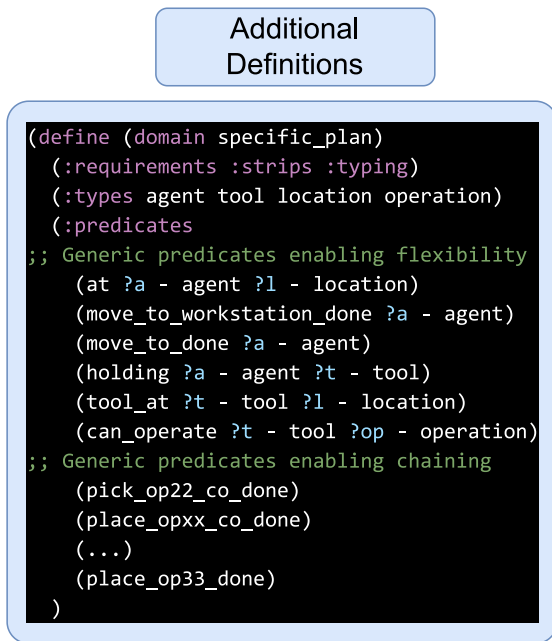


Fig. 6. Excerpt from domain.pddl illustrating the addition of specific predicates necessary for the chaining of operations.

location, a ROS service updates the knowledge base to indicate the absence of the object, and ROSPlan can be triggered to generate a new plan based on current information. The robot adjusts its plan according to the new information from the knowledge base, enabling smoother execution and behavior closer to that of a human in dynamic environments. This establishes a continuous feedback loop between execution outcomes and planning updates, ensuring robustness and adaptivity of the pipeline.

An excerpt of a behavior tree that illustrates the three possible action groups and the decorators to update the PDDL is shown in Fig. 9. The definition of the entities that make up the tree is described in Fig. 10.

### 3.2. Definition of the testing protocol

To test our control structure, we implement it in a scenario involving the Tiago robot. We define a series of actions to be performed by a Human, the Tiago robot, or a Human and Tiago together:

Table 2

List of operations for the test scenario.

Op.	Description	Load (kg)	Height (cm)
$OP_{11}$	Fetch template 1	1	90
$OP_{12}$	Fetch template 2	2	90
$OP_{13}$	Fetch template 3	5	90
$OP_{14}$	Fetch tool 1	1	90
$OP_{15}$	Fetch tool 2	2	90
$OP_{16}$	Fetch tool 3	3	90
$OP_{21}$	Place template 1 on workstation 1	1	90
$OP_{22}$	Place template 2 on workstation 8	2	120
$OP_{23}$	Place template 3 on workstation 6	5	120
$OP_{31}$	Present tool 1 on template 1	1	90
$OP_{32}$	Present tool 2 on template 2	2	120
$OP_{33}$	Present tool 3 on template 3	3	120
$OP_{41}$	Store tool 1	1	90
$OP_{42}$	Store tool 2	2	90
$OP_{43}$	Store tool 3	3	90
$OP_{44}$	Store template 1	1	90
$OP_{45}$	Store template 2	2	120
$OP_{46}$	Store template 3	5	120

The objective is to replicate a case study similar to that described in the work of Hémono et al. (2025), with workstation setup operations  $J_1; J_2$ , transformation operations (schematization of a drilling task by presenting a tool to a fixture)  $J_3$ , and workstation cleanup operations  $J_4$ .

In this scenario, the agents use three drilling templates and three tools (drilling machines) represented by cubes on a vertical panel that simulates the aircraft skin. This panel at the assembly station has two possible rows for use: one at 0.90 m and the other at 1.2 m, as shown in Fig. 11. The tools must be placed in front of the templates and then returned to their initial positions. This practice tests the use of a cobot in a production workshop with human operators performing various tasks. The complete list of operations in the testing protocol, as well as the parameters for the tools and manipulations, is described below in Table 2:

The matrix illustrating the precedence relationships between the operations in Fig. 12 shows that we can intuitively identify two ways to perform the operations. The first method involves parallel execution with three paths identified in blue, green, and orange. The second method is a sequential organization, in which we first handle the setup, then the transformation, and finally the cleanup.

The classification of difficulty levels (WDL) adheres to the guidelines established in our preceding studies (Hémono et al., 2025), which reference a benchmark from an aerospace corporation and are informed by the (Karhu et al., 1977) and AFNOR (2005) standards. We propose hypothetical weight parameters for each tool and template to emulate

Generic Actions

```
;; Generic Action of Movement
(:action move_to
 :parameters (?a - agent ?from - location ?to - location)
 :precondition (at ?a ?from)
 :effect (and
 (at ?a ?to)
 (not (at ?a ?from))
 (move_to_done ?a)
 )
)
```

Fig. 7. Excerpt from domain.pddl illustrating the addition of a generic movement action.

Generated Plan

```
0.001: (move_to agent_r loc_base loc_workstation) [0.001]
0.002: (pick_op22_co agent_r tool_op22_co loc_workstation) [0.001]
0.003: (place_op22_co agent_r tool_op22_co) [0.001]
0.004: (wait agent_r) [0.001]
0.005: (pick_op32 agent_r tool_op32 loc_workstation) [0.001]
0.006: (place_op32 agent_r tool_op32) [0.001]
0.007: (pick_op23_co agent_r tool_op23_co loc_workstation) [0.001]
0.008: (place_op23_co agent_r tool_op23_co) [0.001]
0.009: (pick_op33 agent_r tool_op33 loc_workstation) [0.001]
0.010: (place_op33 agent_r tool_op33) [0.001]
```

Fig. 8. Example of a plan obtained after using the problem interface and planner interface.

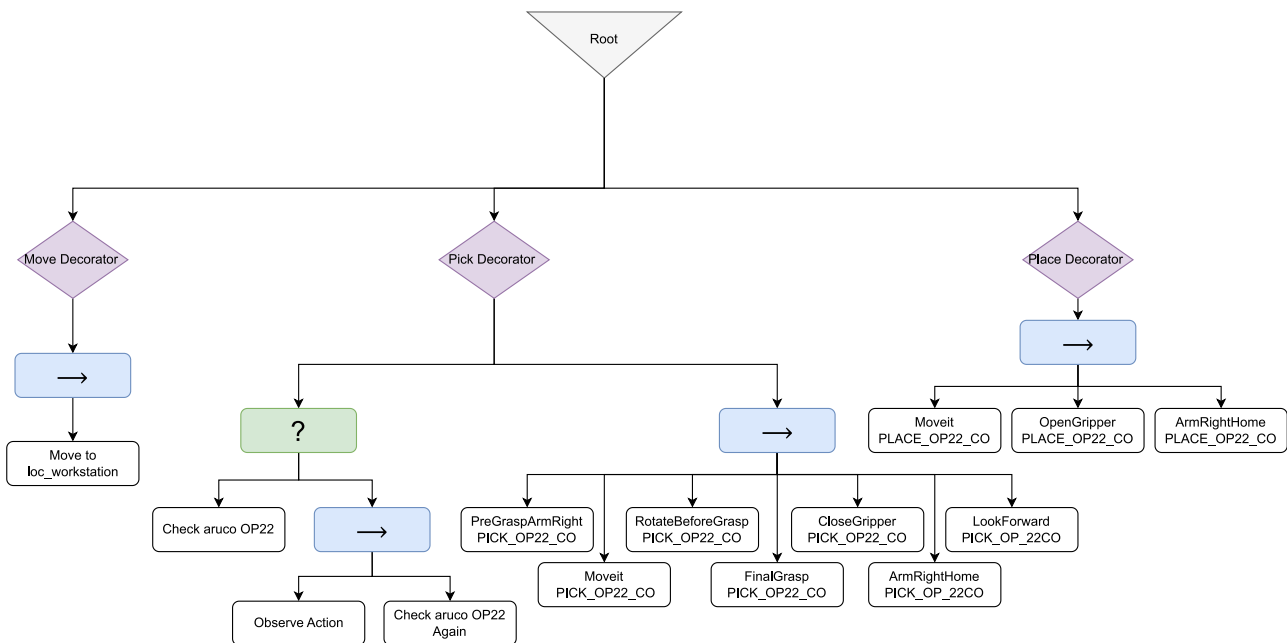


Fig. 9. Example of a Behavior Tree obtained with Assembler P.3.

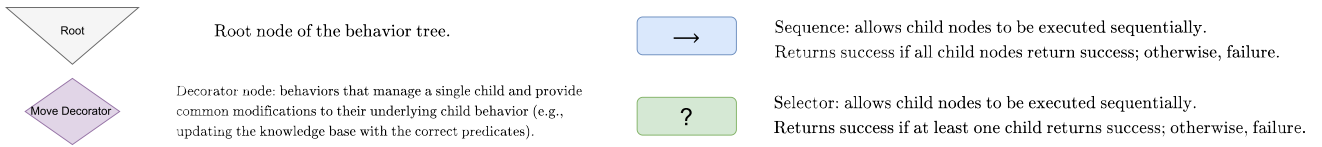


Fig. 10. Definition of Behavior Tree Entities.

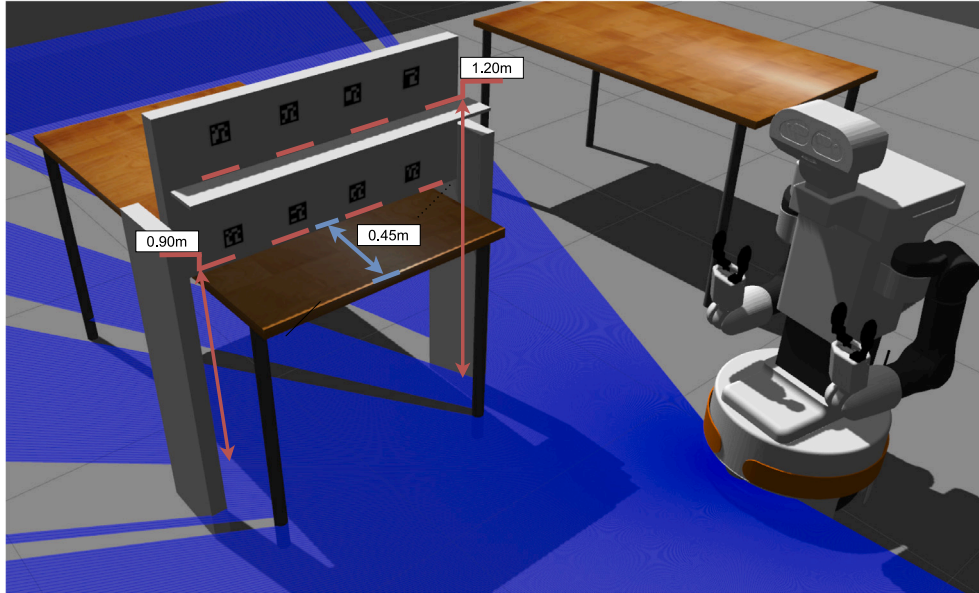


Fig. 11. Simulation of work environment.

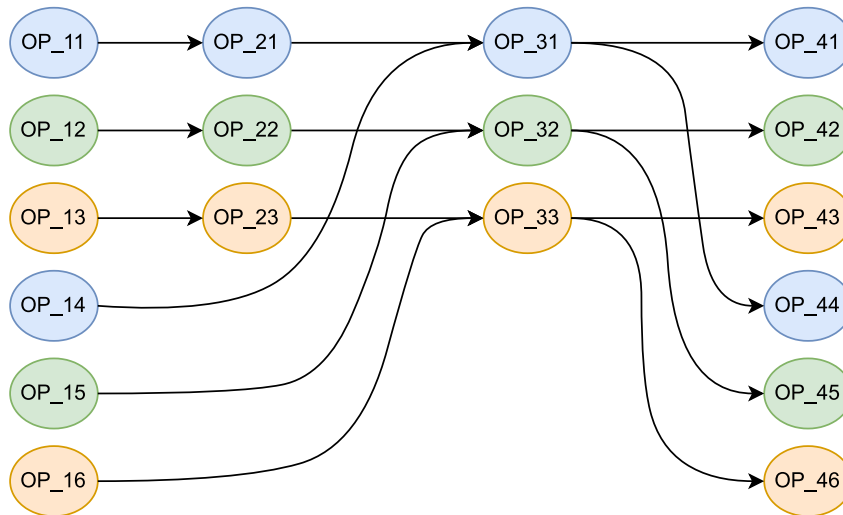


Fig. 12. Precedence matrix of our test scenario.

diverse levels of difficulty and effort required for different tasks. The corresponding values are detailed in Table 2.

3.2.1. Definition of difficulty levels for the human agent:

In our scenario, there are no situations where bending or knee flexion is necessary; therefore, the corresponding WDL levels are set to 1 for all operations.

In the test environment, there are 8 work zones. Zones 1 to 4 are at a height of 0.90 m and a distance of 45 cm from the operator's body axis while using the tool or template. In contrast, work zones 5 to 8 are at a height of 1.2 m and a distance of 45 cm from the

body axis. For a person measuring 1.75 m, working at a height of 90 cm positions the table at elbow height, thereby meeting ergonomic specifications for standing work at a table. The associated WDL level is 1 (frequent angulation < 20°). However, for the same individual, working at a height of 1.2 m requires frequent shoulder angulation between 60° and 90°, corresponding to a WDL level of 4 (exceeding ergonomic guidelines in terms of limits and duration).

The weights of the tools and templates are all within the gripping capacity of the agent. For template 3, which weighs 5 kg, handling increases the WDL to 2 (effort/load between 5 kg and 10 kg (frequent or occasional), and < 5 kg (frequent)). The WDL level for the other cubes is 1 for efforts and loads.

**Table 3**  
Difficulty parameters ( $W_{ijk}$ ) and duration parameters ( $t_{ijk}$ ) in second for task execution.

Operation	$W_{ijH}$	$W_{ijR}$	$W_{ijCo}$	$t_{ijH}$	$t_{ijR}$	$t_{ijCo}$
$OP_{11}$	1	2	1	10	60	50
$OP_{12}$	1	2	1	10	60	50
$OP_{13}$	2	4	3	10	60	50
$OP_{14}$	1	2	1	10	60	50
$OP_{15}$	1	2	1	10	60	50
$OP_{16}$	1	2	1	10	60	50
$OP_{21}$	1	2	1	5	45	8
$OP_{22}$	2	2	1	5	45	8
$OP_{23}$	4	2	1	5	45	8
$OP_{31}$	1	1	1	5	45	8
$OP_{32}$	2	1	1	5	45	8
$OP_{33}$	4	1	1	5	45	8
$OP_{41}$	1	2	1	10	60	50
$OP_{42}$	1	2	1	10	60	50
$OP_{43}$	1	2	1	10	60	50
$OP_{44}$	1	2	1	10	60	50
$OP_{45}$	1	2	1	10	60	50
$OP_{46}$	2	4	3	10	60	50

For combined postural efforts and load handling, the WDL level considered is the highest encountered during the operation. A summary of difficulty levels is presented in Table 3.

### 3.2.2. Definition of difficulty levels for the Tiago Robot:

The robotic agent remains unaffected by repetitive movements or load handling, provided these operations fall within its mechanical capacity limits. In the environment, all test zones are accessible to the Tiago Robot. The primary difficulty lies in the precision required to detect objects to manipulate and in the robot's movement capability. For the simulated transformation operations, high precision is not necessary since it is sufficient to position the cube in front of the one representing the template, so the WDL level for these operations is set to 1.

We assigned a WDL level of 2 for the precise handling of a cube, including its grasping and placement. However, if the weight of the cube exceeds the maximum capacity of the robotic arm, the difficulty level for the gripping sequence is evaluated at 4.

### 3.2.3. Definition of difficulty levels for collaborative mode:

In collaborative mode, human and robotic capabilities are combined to maximize their respective strengths. Handling operations become easier for the human, as they are exempted from lifting heavy loads, while the robot overcomes challenges related to detection and precision. The collaborative grasping task ( $OP_{11-16}$  and  $OP_{41-46}$ ) is performed as follows: Tiago's arm switches to gravity compensation mode, and the human guides the movement to position the gripper on the cube and close it. To place the cube, the reverse movement is performed. For transformation operations (drilling) ( $OP_{21-23}$  and  $OP_{31-33}$ ), the robot holds the tool, and the robotic arm switches to gravity compensation mode while the human guides the movement. Thus, the weight of tools and templates is supported by the robot, reducing the human stress of lifting a load, repetitive movements, and working in elevated zones. The WDL level for handling in collaborative mode is defined as 1, except for the operation involving template 3, which requires both of Tiago's arms; in this case, the WDL level is set to 3. The difficulty levels for each agent are summarized in Table 3.

### 3.2.4. Definition of operation durations:

The execution durations for each operation performed by the human agent, the robot, or collaboratively by the human and robot together are measured during tests carried out to complete the full scenario. The values for the duration of the operation are presented in Table 3.

**Table 4**  
Data extracted and reprocessed by the converter.

Operation	Job	Resource	Start Time ( $S_{ijk}$ )	End Time ( $C_{ijk}$ )
$op_2$	$j_1$	H	0	10
$op_2$	$j_2$	Co	10	18
$op_5$	$j_1$	H	18	28
$op_2$	$j_3$	R	28	73
$op_4$	$j_1$	H	28	38
$op_1$	$j_1$	H	38	48
$op_3$	$j_1$	H	48	58
$op_6$	$j_1$	H	58	68
$op_1$	$j_2$	H	68	73
$op_3$	$j_2$	Co	73	81
$op_1$	$j_3$	H	81	86
$op_3$	$j_3$	R	81	126
$op_1$	$j_4$	H	86	96
$op_4$	$j_4$	H	96	106
$op_2$	$j_4$	H	106	116
$op_5$	$j_4$	H	116	126
$op_3$	$j_4$	H	126	136
$op_6$	$j_4$	H	136	146

### 3.3. Results and discussions

Once the study parameters are well defined, the approach consists of implementing the three main modules of the pipeline defined in Fig. 4.

The first step involves optimizing the problem with Gurobi, aiming to reduce the strain on the human agent and the overall execution time of the task, as described in Hémono et al. (2025). For optimization, we selected an  $\alpha$  value of 0.45, which, as previously noted, represents a good balance between the two optimization objectives.

The optimization results produce the following Gantt diagram 13. By examining the difficulty level of the tasks for agents, a comparison between the process performed solely by the human and the process in which the human works in synergy with the robot shows a reduction in operator strain from  $W_{max} = 28$  to  $W_{max} = 20$ , representing a reduction of 28.57%. The execution time decreases from 150 s for the human alone to 146 s with the robot, a 2.67% reduction, due to the parallel execution of certain tasks by the robot.

It should be noted that operations  $OP_{22}$  and  $OP_{23}$  are performed in collaborative mode. These operations involve handling templates 2 and 3, weighting 2 kg and 5 kg respectively, and placing them in work zones at a height of 1.20 m, as illustrated in Fig. 11. Performing these operations in collaborative mode is particularly beneficial, as it reduces the difficulty level to 1 for both operations, despite the additional time required for their execution.

The allocation of tasks  $OP_{32}$  and  $OP_{33}$  to the robot relieves the human agent of a non-ergonomic manipulation task without introducing any waiting time, thus reducing the total process execution time. These results assess the results and insights given in Hémono et al. (2025).

After optimization, the next step in the pipeline is filtering the optimization logs using the Converter P.1. The processed and filtered log file is shown in Table 4. The extracted and filtered data aligns well with the Gurobi plan presented in Fig. 13. Subsequently, the file of elementary operations must be manually generated, which, along with the filtered logs, serves as input for Dispatcher P.2. This file is in JSON format and records the primary nature of each action as "Pick," "Place," or "Move to." The complete file generated for our specific case is provided in Github.

The next step of our pipeline involves using the optimization logs along with the file of elementary operations as input for the Dispatcher to generate the PDDL knowledge base. The proposed plan, based on the files presented earlier, is shown in our github repository. It is observed that the Dispatcher successfully generates a domain.pddl file that accurately corresponds to the actions the robot must perform either alone or in collaboration. In our case, the operations are:  $OP_{22}$  in

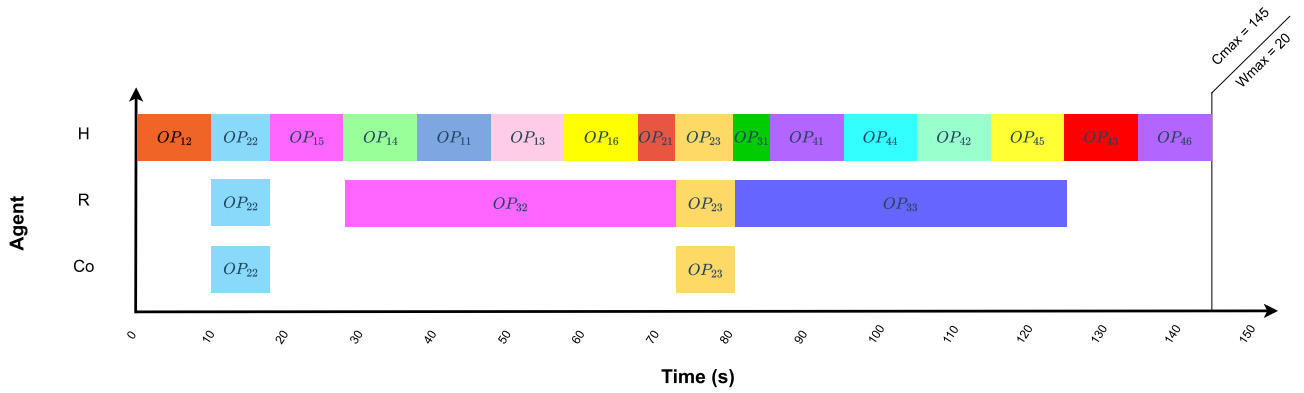


Fig. 13. Gantt chart obtained for our case study with  $\alpha = 0.45$  and the parameters defined in Table 3. H: Human agent; R: Robot Agent; Co: Human and robot agent together.

collaborative mode,  $OP_{32}$  alone,  $OP_{23}$  in collaborative mode, and then  $OP_{33}$  alone. The plan calculated from the knowledge base matches the one proposed in Fig. 8. The operations are correctly transcribed in the plan and arranged in the initially planned order.

The Dispatcher also integrates a wait function into the `domain.pddl` when the end time of operation  $n-1$  does not coincide with the start time of operation  $n$ . This prevents the robot from starting an action before other agents have completed the previous operations.

The next step is to construct the behavior tree using the Assembler from our pipeline. The tree will be built following the logic presented in Section 3.1 and partially illustrated in Fig. 9. The complete initial behavior tree is shown in Fig. B.23 of Appendix B.

During the execution of the behavior tree, any unforeseen events that may occur will allow, through the decorators, an update of the knowledge base. If a final state is reached, a new computation is performed to adjust the task plan in order to achieve the original objectives.

We intentionally test a failure scenario to evaluate the resilience of our model. In the initial operations, Tiago must move to the workstation to fetch a template that should have been brought to the station during the previous operation performed by the human agent. Tiago must then grasp this template in collaborative mode to position it on the panel at its designated location.

We deliberately return the template to its original location to simulate an operator's oversight in completing this action and to observe the behavior of our pipeline.

The new information on the successive states reached is added to the knowledge base through the decorators, as shown in Fig. 15:

The robot successfully completes its movement and acquires a new location. This is indeed shown in Fig. 14, where it is seen at the workstation observing to locate the template, which is not present at the station.

In a subsequent step, updates are added to the knowledge base regarding the position of the tool `tool_op22`, which is not where it is supposed to be. In our case study, there are two possible locations for the tools: the workstation and the storage station. If the tool is not at one of these two stations, we deviate from the normal operating schema. Therefore, we systematically consider the tool's position at the opposite station (if it is not detected) as the only possible alternative in our specific case. Of course, the logic we use can be adapted to define other strategies depending on the scenarios.

The mention `tool_at loc_base` is thus added since the tool is not detected at `loc_workstation`. The updated problem.pddl is presented in our [github repository](#).

When a failure state is reached in our behavior tree, we generate and resolve a new instance of the problem to design a new plan adapted to the objective. This dynamic management helps the robot recover from its state of uncertainty or failure. Additionally, an alert strategy can

be included, where the robot signals verbally or using a light indicator that it has reached a state of failure or uncertainty. This prevents the generation of a new plan if the human agent can correct the issue in a simple and direct manner.

The new plan is calculated and shown in Fig. 16.

As observed in Fig. 16, the solver has automatically added two actions from the available generic actions. This allows the robot to move to the new location of the tool required for operation  $OP_{22}$ , grasp it, and return it to the workstation. The rest of the plan remains identical to the one initially generated.

This degraded plan enables the robot to recover from its blocked state without disrupting the original organization or the task distribution between the human and the robot. The new plan in execution is shown in Fig. 17.

Due to the update of the knowledge base, the continuation of plan execution is made possible. Fig. 18 shows that the rescheduling has integrated two additional actions for the robot, which planned the necessary generic movements to compensate for the gap caused by the failure or incomplete execution of  $OP_{12}$  by the human agent.

In Fig. 18, it is observed that operation  $OP_{12}$ , indicated by cross-hatching, was not performed, resulting in failure of  $OP_{22}$ , shown with simple hatching. This leads to an increase in the makespan of 13.10% compared to the optimal plan, rising from 145 to 164 s, and an increase of 9.33% compared to the initial task execution plan.

Despite this time degradation, it is important to note that this ensures the continuity of actions and maintains a significant reduction in WDL by 28.57%. Without this continuity, the robot could remain blocked for an indefinite period, thus worsening the effective task execution time compared to the ideal scheduling. We also note a waiting time of 10 s for the human agent, which in general should be avoided to prevent operator frustration.

This waiting time occurs when the optimal plan is degraded to ensure operational continuity. For instance, the digital twin can identify a discrepancy such as a tool not being where it was expected potentially due to human error, such as the operator forgetting to return the tool to the workstation. In such cases, the robot may initially be blocked. However, leveraging the pipeline's adaptive capabilities, the system can generate an alternative plan to recover the missing tool from the storage station before resuming the original sequence of activities.

At this point, the robot communicates its intentions to the human operator using signaling methods such as a display or visual cues (e.g., lights). The human operator is presented with two options: acknowledge the oversight and opt to manually rectify the issue — preventing the robot from executing the new plan or remain passive, allowing the robot to autonomously complete the recovery and continue with the updated plan.

Future work on system resilience could explore re-optimizing degraded plans using Gurobi's multi-objective optimization. Specifically,

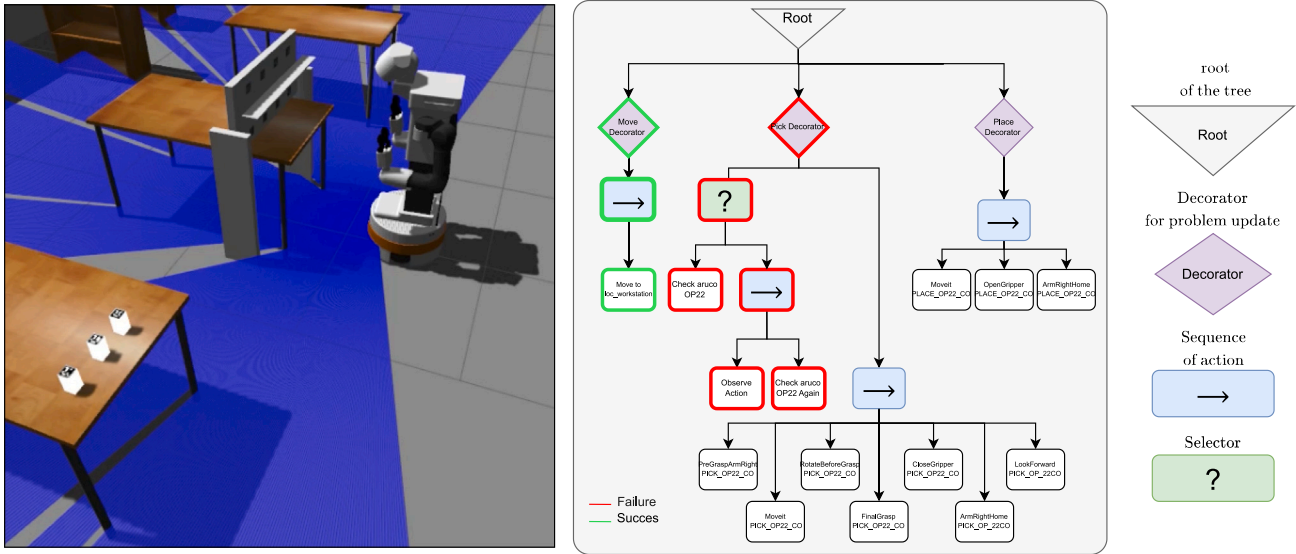


Fig. 14. The Tiago robot successfully completes the first movement action with a "success" state but fails to detect the template and returns a "failure" state.

problem.pddl  
Update

```
[INFO] KCL: (/rosplan_planner_interface) Adding Fact
(at agent_r loc_workstation, 0)
[INFO] KCL: (/rosplan_planner_interface) Adding Fact
(move_to_loc_workstation_done, 0)
[INFO] KCL: (/rosplan_planner_interface) Removing Fact
(at, 0)
[INFO] KCL: (/rosplan_planner_interface) Removing Fact
(tool_at, 0)
[INFO] KCL: (/rosplan_planner_interface) Adding Fact
(tool_at tool_op22_co loc_base, 0)
```

Fig. 15. Excerpt from the ROSPLAN terminal with knowledge base update.

New Plan

```
■ move_to agent_r loc_workstation loc_base
■ pick_op22_co agent_r tool_op22_co loc_base
■ move_to agent_r loc_base loc_workstation
■ place_op22_co agent_r tool_op22_co
■ wait agent_r
■ pick_op32 agent_r tool_op32 loc_workstation
■ place_op32 agent_r tool_op32
■ pick_op23_co agent_r tool_op23_co loc_workstation
■ place_op23_co agent_r tool_op23_co
■ pick_op33 agent_r tool_op33 loc_workstation
■ place_op33 agent_r tool_op33
```

Fig. 16. New plan calculated based on the updated problem.pddl during the execution of the behavior tree.



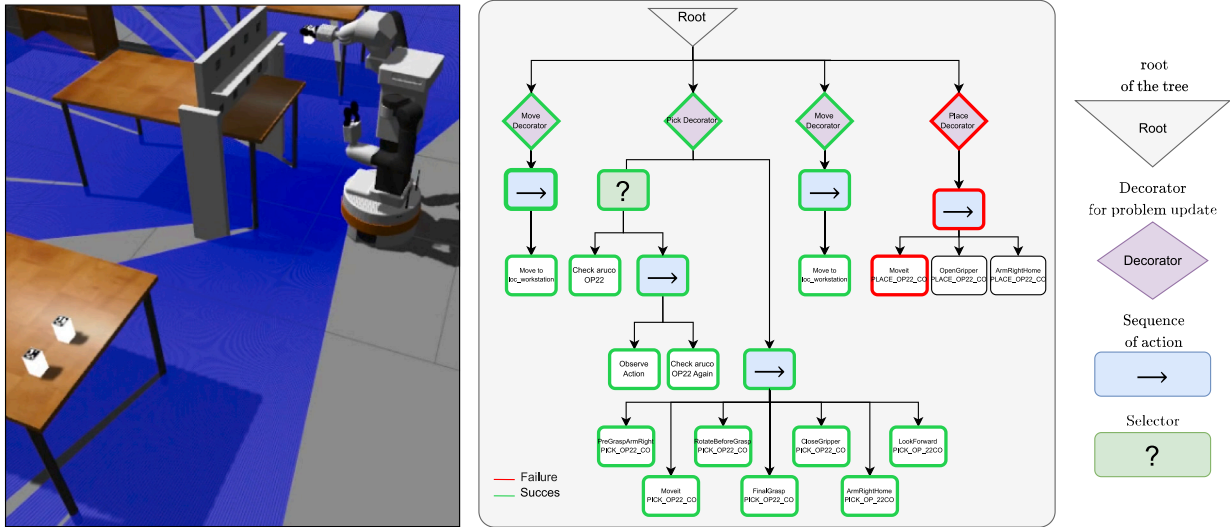


Fig. 19. Execution of the new plan until the appearance of the new Failure state.

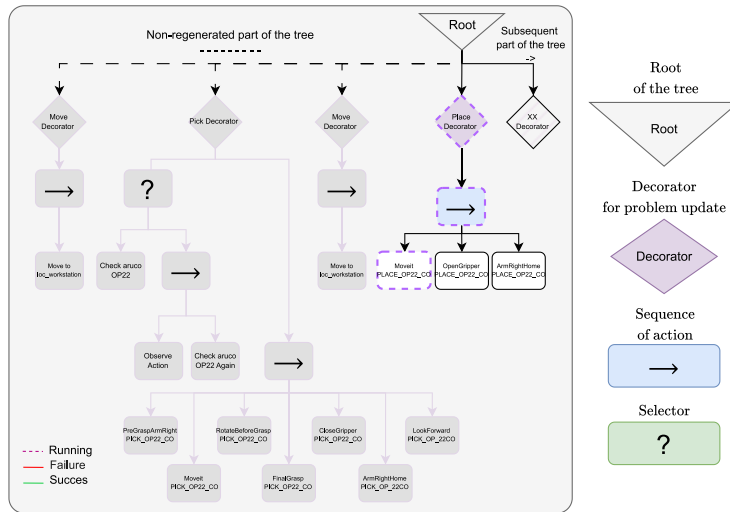


Fig. 20. New Behavior Tree. Actions in gray are not regenerated by the Pipeline Assembler P.3.

4. Managerial insight

The integration of Digital Twin technology with dynamic scheduling marks a significant advancement in enhancing the resilience and adaptability of industrial operations. By dynamically adjusting to unexpected disruptions, such as human mistakes or unplanned occurrences, this method ensures continuous operations without significant interruptions. Notably, recalculating task allocations whenever an incident arises is not always necessary, as this may lead to confusion and require the operator to pause in order to understand the updated plan, potentially delaying execution. A two-tiered strategy allows for the formulation of a suboptimal plan that sustains operational flow, with the ultimate goal of staying aligned with the optimal plan. For example, if a tool is absent from its anticipated location due to human oversight, the system autonomously creates an alternative recovery plan, allowing the robot to retrieve the missing resource and then proceed with the original activities. This flexibility minimizes bottlenecks and enhances the overall robustness of the production process.

Unlike approaches relying on Digital Human Modeling, which often require continuous observation of the human operator, our framework

focuses on optimizing the dynamic management of tasks performed by robots. Digital Human Modeling, while valuable, can inadvertently generate stress, discomfort, or mistrust among workers. Monitoring movements often requires wearable devices such as exoskeletons, sensors, or augmented reality glasses, which can be a burden over long periods. In contrast, our approach avoids imposing additional physical or mental strain on human operators. Instead, we prioritize designing intelligent processes and enhancing robotic capabilities to ensure that the robot’s actions remain both effective and intuitively comprehensible, yet almost invisible in their execution.

This paradoxical but deliberate human-centered approach emphasizes improving task sequencing and robotic adaptability to seamlessly support workers. Robots are thus enabled to dynamically manage unexpected situations, such as task disruptions, without requiring intrusive monitoring of the human agent. As a result, this method preserves the well-being of workers while maintaining productivity and operational flexibility.

In addition, this framework fosters trust and transparency in human-robot collaboration by allowing clear communication of the robot’s intentions through visual displays or signals. Human operators are

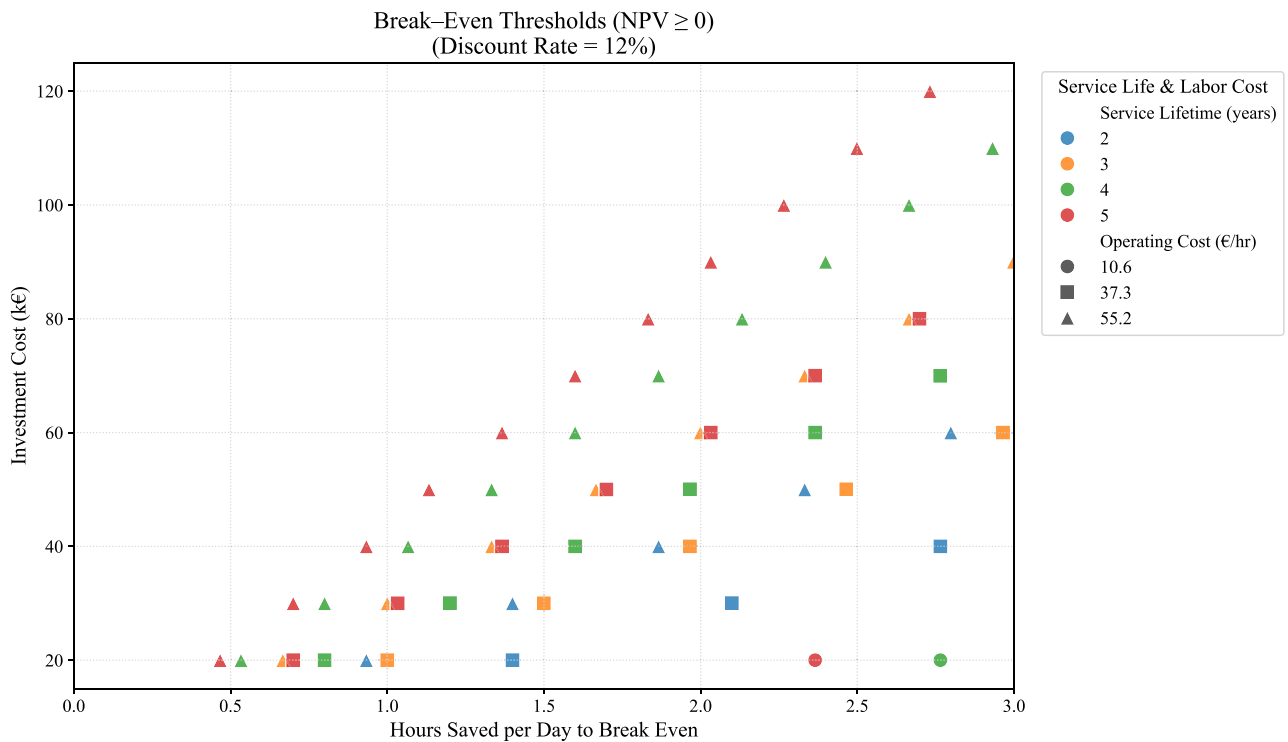


Fig. 21. Break-even thresholds (NPV  $\geq$  0).

empowered to decide whether to intervene manually or allow the robot to proceed autonomously. From a managerial perspective, the proposed solution combines high-level optimization (e.g., Gurobi-based multi-objective optimization) with real-time recovery mechanisms, ensuring that tasks are executed efficiently and collaboratively while maintaining resilience to operational disruptions. This positions organizations to leverage the benefits of Industry 5.0 by creating sustainable, adaptive, and human-centric production systems that enhance collaboration between human agents and robotic systems.

#### 4.1. Techno-economic evaluation: Managerial considerations

The enhanced resilience of digitally-twin-enabled operations can translate into significant cost savings over time. To quantify this potential, we draw inspiration from the techno-economic methodology of Choux et al. (2024), which combines capital cost annuitization with multi-year present value analysis. This framework delivers decision-grade metrics annualized capital cost, Net Present Value (NPV), and Return on Investment (ROI) under varying efficiency gains.

From a managerial standpoint, evaluating the adoption of mobile manipulators through a techno-economic lens reveals key insights that directly inform investment decisions. Several operational and economic levers critically affect the profitability and payback of such robotic systems.

Fig. 21 provides a break-even analysis focusing on daily hours saved in relation to investment costs. To highlight patterns and improve understanding, profitability is evaluated at three different labor rates: a minimum of €10.6/h, the least expensive labor rate in Europe for 2024, an average rate of €37.3/h, the mean European labor cost, and a maximum of €55.2/h, the highest rate among European nations in 2024, based on Eurostat (2024).

Fig. 21 also demonstrates how variations in investment, labor rates, and amortization periods affect daily time-saving needs. At the highest rate (€55.2/h), an extended payback period of five years for an initial investment of €100 k necessitates only about  $\approx$ 2.3 h of daily productivity enhancements led by robotics. Reducing the investment

amount to €60 k or €20 k correspondingly decreases this threshold to approximately  $\approx$ 1.4 and  $\approx$ 0.4 h/day, respectively. However, if the emphasis is on achieving a swift two-year payback, these minimums increase significantly to  $\approx$ 2.7 h/day for an investment of €60 k, and  $\approx$ 0.9 h/day for €20 k. Notably, there is no feasible scenario for realizing 3 h of daily savings with an investment of €100 k within the specified time frame.

For a labor rate of €37.3/h, within a five-year span, the break-even point is not reached with an initial investment of €100k. However, reducing the investment to €60k or €20k lowers the necessary productivity to  $\approx$ 2.1 h/day and  $\approx$ 0.7 h/day, respectively. When the time frame is shortened to two years, the required productivity savings increase to approximately  $\approx$ 1.4 h/day for an investment of €20k, emphasizing how both labor costs and the duration of payoff influence productivity requirements. Investments of €100k and €60k fail to achieve profitability within this observed two-year period for a time window of 3 h.

At the lowest labor cost (€10.6/h), only the longest payback schedules fall within our 0–3 h/day window. For a €20 k investment, a four-year horizon requires roughly  $\approx$ 2.7 h/day of robotic time-savings to break even, and a five-year horizon about  $\approx$ 2.4 h/day. In contrast, neither the two nor three-year schedules ever reach positive NPV before 3 h/day, and investments of €60 k or €100 k likewise fail to intersect the profit threshold within three hours of daily savings for any amortization term. Thus, in low-wage settings, only extended amortization periods or lower investments costs can bring required productivity gains into a realistic range.

Taken together, these interpolated break-even points make explicit that reducing the initial investment delivers an almost linear reduction in daily hours-saved requirements, particularly at high wage rates. In mid-tier scenarios, projects under €60 k become accessible with under 2 h/day of gains over five years versus 3 h/day for €100 k. And even in low-cost settings, a minimal €20 k outlay can break even with just 1.4 h/day of productivity improvements across a five-year horizon underscoring the central role of amortization in spreading capital expenditures across the asset's productive life. These quantifications equip managers to calibrate human-robot productivity objectives precisely against both labor rates and realistic payback expectations.

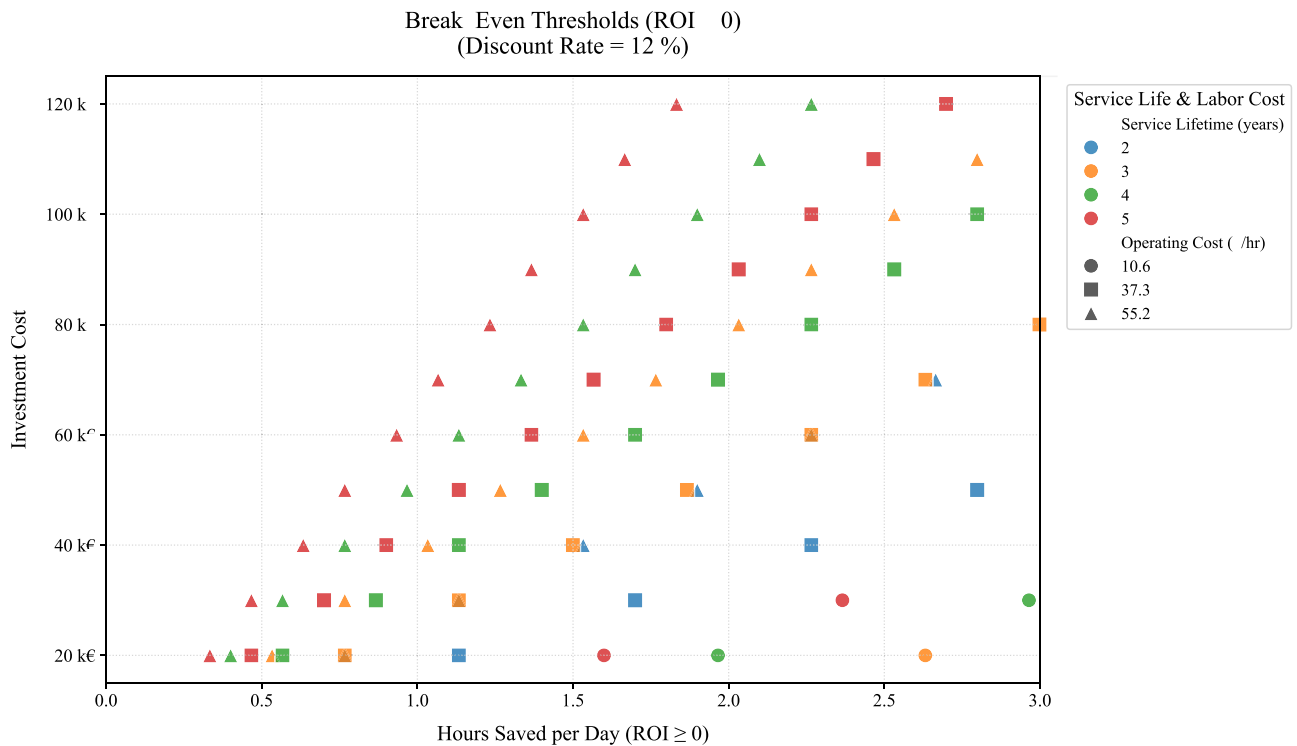


Fig. 22. Break-even thresholds (Return On Investment  $\geq 0$ ).

In practice, true downtime often exceeds these hourly thresholds. A lights-out production cell running nights and weekends can incur 48 h of idle time in a single week. Five such events per year average 1 h/day of lost production well above most break-even levels (0.6–1.7 h/day). Emergency call-out of technicians adds premium labor and logistics costs, and delayed shipments risk penalties or lost contracts whose financial impact dwarfs our hourly savings targets.

Given the important costs per unit in sectors such as aerospace or semiconductor fabrication, even a single unplanned stoppage can translate into million-euro losses. Penalties for late delivery, rework of defective components, or contract terminations all underscore the need for a production framework that is not only efficient, but truly resilient—capable of anticipating disruptions, re-planning on-the-fly, and keeping high-value lines running without interruption.

Our dynamic-scheduling approach directly mitigates these risks. Rather than freeze when a tool is missing or a process changes, the robot's digital twin automatically re-plans—selecting alternate resources, reordering tasks, and maintaining progress without human dispatch. This agility reduces effective downtime, smooths throughput, and accelerates the path to positive NPV.

The Fig. 21 illustrates that reducing investment costs accelerates the break-even point. The prospect of a robot capable of conducting pick-and-place operations and maneuvering within a workshop is increasingly feasible with advancements like Tesla's Optimus robot priced at approximately €26000 or Unitree Robotics' G1 at around €14000. In nations with high labor costs, the returns on investment and benefits from such solutions could significantly increase.

Fig. 22 recasts the same break-even contours in terms of ROI=0 instead of NPV=0. Because ROI simply measures cumulative cash return vs. capital outlay, its zero-line is reached at fewer daily savings hours than the NPV criterion, especially for medium and long service lives.

In practical terms, ROI can serve as a more intuitive metric for line managers, who often reason in terms of "years to recoup investment" rather than relying on net present value (NPV). It emphasizes that

given long service lifespans and high labor costs—even modest daily savings can yield a 100% ROI well within the operational lifetime of the equipment. Furthermore, since ROI does not involve discounting beyond simple payback, it can more directly facilitate capital budgeting discussions. For instance: "If we invest €100k today, this robotic line will pay for itself in  $\leq 5$  years, provided we save just 1 h of labor per day.

Taken together, the NPV and ROI analyses provide a consistent narrative: capital investments amortized over longer time horizons—especially under elevated labor cost scenarios—significantly reduce the time required for automation systems to break even. Consequently, managers are advised to prioritize high-cost, continuous-operation production lines (e.g., 24/7 systems, where actual downtime often exceeds 1 h per day) and to engage in multi-year service commitments in order to maximize returns in terms of both NPV and ROI.

## 5. Discussion and theoretical implications

This study set out to explore how digital twins (DTs) and dynamic scheduling can enhance human-robot interaction, to identify missing human-centric control mechanisms in the DT literature, and to investigate the role of behavior trees (BTs) in improving adaptability and transparency.

Most DT frameworks excel at data synchronization and visualization but rarely incorporate the decision logic required for truly adaptive operation. By embedding our dual-layer dynamic scheduler, the twin becomes an active coordinator: an offline multi-objective optimizer defines a strategic reference plan, while a lightweight, BT-driven replanning engine responds instantaneously to disturbances. This closed-loop design preserves throughput and prevents cognitive overload from constant micro-adjustments.

We further introduce a memory-driven mechanism that records and reuses past recovery patterns. This experiential loop not only accelerates response to recurring events but also addresses the absence of historical context retention in existing DT studies. At the same

time, behavior trees structure each replanning step into clear, modular decision stages, making adaptations transparent and enabling operators to trace and, when necessary, override specific actions.

Importantly, our scheduler modulates replanning frequency and granularity to align with human cognitive bandwidth. By tailoring updates to operator needs, the system enhances collaboration and reduces misunderstandings—an aspect often overlooked in techno-centric DT models.

Compared to approaches that ignore operator cognition or lack real-time control, our framework demonstrates measurable gains in responsiveness and user trust. While preliminary simulations confirm these benefits, future work will involve user studies in industrial settings to quantify operator workload and validate long-term system learnings.

In summary, embedding a closed-loop dynamic scheduler with memory-aware experiential learning and BT-based execution transforms digital twins into proactive, human-oriented agents. This theoretical contribution advances DTs from passive replicas to intelligent collaborators capable of resilient, adaptive operation in Industry 5.0 environments.

## 6. Conclusion and future works

In this research, we explored a dual layer framework that combines high-level scheduling through optimization techniques and real-time robotic task execution via Behavior Trees (BT), integrated into a Digital Twin (DT) platform. Our approach demonstrated improvements in human-robot collaboration by reducing operator workload and optimizing task execution in dynamic environments. This methodology is aligned with the principles of Industry 5.0, prioritizing human-centric design, flexibility, and operational resilience. Furthermore, we demonstrate that increased resilience can positively impact production economics, possibly facilitating the broader adoption of this type of framework.

The integration of a Digital Twin allows for real-time monitoring and dynamic plan adjustments, enhancing the adaptability of robotic systems. Behavior Trees, with their modular and hierarchical design, facilitate seamless transitions between predefined and reactive behaviors, thereby supporting human-robot interaction and trust-building. The pipeline demonstrated the capacity to recover from unforeseen events, maintain task continuity, and mitigate operational disruptions.

Our methodology has limitations requiring attention to maximize the dual-level framework's potential. Current validation is limited to a structured indoor scenario, so broader assessments are needed. The pre-defined behavior trees can cause deadlocks requiring human intervention, thus, AI-enhanced Behavior Trees are recommended for dynamic branching, improving resilience and reducing external aid. While suited for industrial planning, more complex areas like home assistance and healthcare may need enhanced learning and context awareness for similar autonomy and robustness. Our system assumes consistent Gurobi plan execution but does not consider uncertainties like human refusal or delays, or plan downgrades due to alarms, which require defined criteria for re-optimizing plans during failures.

### Future perspectives

- **Human-Centered Enhancements:** Future research should deepen the integration of human factors, including cognitive aspects, to further align robotic actions with operator needs. This includes minimizing monitoring stress by emphasizing task adaptability rather than invasive human modeling.
- **Advanced Optimization Models:** Incorporating multi-objective optimization frameworks like Gurobi's capabilities could address plan degradation thresholds, offering adaptive solutions when performance metrics such as workload levels or makespan deviate significantly from optimal values.

- **Ethical and Trust Considerations:** Addressing the psychological impacts of collaborative systems, such as transparency and user autonomy, can promote greater acceptance. This includes implementing non-intrusive signaling mechanisms and refining robot communication methods.
- **Resilience Mechanisms:** Enhancing the pipeline to include predictive analytics and self-learning mechanisms could further strengthen resilience, allowing preemptive adjustments based on anticipated disruptions or operator feedback.

This research lays the groundwork for future advancements in human-robot collaboration, emphasizing a balanced integration of technological innovation and human-centric design. By refining the interplay between dynamic scheduling and robotic execution, we move closer to realizing the vision of resilient and sustainable Industry 5.0 ecosystems.

### CRedit authorship contribution statement

**Pierre Hémono:** Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Ahmed Nait Chabane:** Writing – review & editing, Writing – original draft, Validation, Supervision, Project administration, Methodology, Formal analysis, Conceptualization. **M'hammed Sahnoun:** Writing – review & editing, Writing – original draft, Validation, Supervision, Project administration, Methodology, Investigation, Formal analysis, Conceptualization.

### Acknowledgments

The authors acknowledge the financial support of ANRT (National Association of Research and Technology) for funding this research, No. 2021/0509. They also thank eXcent company for supporting this work. This work has also benefited from the institutional support of the University of Agder (UiA), Norway.

### Appendix A. Pseudo-code for the complete pipeline

*Block 1: Optimization 5.0 (includes P.1 Converter and P.2 Dispatcher)*

```

1 def optimization_5_0(solution_file,
2   elementary_operations_file):
3     # Step 1: Convert Gurobi logs with P.1
4     Converter
5     solution_readable =
6     p1_converter(solution_file)
7
8     # Step 2: Generate PDDL files with P.2
9     Dispatcher
10    domain_pddl, problem_pddl =
11    p2_dispatcher(solution_readable,
12    elementary_operations_file)
13
14    return domain_pddl, problem_pddl
15
16 def p1_converter(solution_file):
17     # Read and process the solution file
18     solution_readable =
19     convert_gurobi_logs(solution_file)
20     return solution_readable
21
22 def p2_dispatcher(solution_readable,
23   elementary_operations_file):
24     # Load elementary operations and
25     converted logs

```



## References

- Abiyev, R.H., Günsel, I., Akkaya, N., Aytac, E., Çağman, A., Abizada, S., 2016. Robot soccer control using behaviour trees and fuzzy logic. *Procedia Comput. Sci.* 102, 477–484. <http://dx.doi.org/10.1016/j.procs.2016.09.430>.
- AFNR, 2005. Safety of machinery - human physical performance - part 4: Evaluation of working postures and movements in relation to machinery. Technical Standard, nF EN 1005-4.
- Agis, R.A., Gottifredi, S., García, A.J., 2020. An event-driven behavior trees extension to facilitate non-player multi-agent coordination in video games. *Expert Syst. Appl.* 155, 113457.
- Aloulou, M.A., Portmann, M.-C., 2005. An efficient proactive-reactive scheduling approach to hedge against shop floor disturbances. In: *Multidisciplinary Scheduling: Theory and Applications: 1 St International Conference, MISTA'03 Nottingham, UK, 13–15 2003 Selected Papers*. Springer, pp. 223–246.
- Arisoy, E.B., Ren, G., Ulu, E., Ulu, N.G., Musuvathy, S., 2016. A data-driven approach to predict hand positions for two-hand grasps of industrial objects. In: *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 50077, American Society of Mechanical Engineers, V01AT02A067.
- Banerjee, B., 2018. Autonomous acquisition of behavior trees for robot control. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, IEEE*, pp. 3460–3467. <http://dx.doi.org/10.1109/iro.2018.8594083>.
- Bányai, T., Kovács, G., 2023. Identification of influence of digital twin technologies on production systems: a return on investment-based approach. *East-European J. Enterp. Technol.* 6 (3), 49–59. <http://dx.doi.org/10.15587/1729-4061.2023.283876>.
- Bécue, A., Maia, E., Feeken, L., Borchers, P., Praça, I., 2020. A new concept of digital twin supporting optimization and resilience of factories of the future. *Appl. Sci.* 10 (13), 4482.
- Berti, N., Finco, S., 2022. Digital twin and human factors in manufacturing and logistics systems: state of the art and future research directions. *IFAC-PapersOnLine* 55 (10), 1893–1898.
- Bilberg, A., Malik, A.A., 2019. Digital twin driven human–robot collaborative assembly. *CIRP Ann* 68 (1), 499–502.
- Bouaziz, N., Bettayeb, B., Sahnoun, M., Yassine, A., 2024. Incorporating uncertain human behavior in production scheduling for enhanced productivity in industry 5.0 context. *Int. J. Prod. Econ.* 274, 109311.
- Chen, M., Nikolaidis, S., Soh, H., Hsu, D., Srinivasa, S., 2020. Trust-aware decision making for human–robot collaboration: Model learning and planning. *ACM Trans. Human-Robot Interact. (THRI)* 9 (2), 1–23.
- Child, C., Dey, R., 2013. Ql-bt: Enhancing behaviour tree design and implementation with q-learning. In: *Computational Intelligence in Games (CIG), 2013 IEEE Conference on, IEEE*, pp. 275–282. <http://dx.doi.org/10.1109/cig.2013.6633623>.
- Choux, M., Marti Bigorra, E., Tyapin, I., 2021. Task planner for robotic disassembly of electric vehicle battery pack. *Metals* 11 (3), 387.
- Choux, M., Pripp, S.W., Kvalnes, F., Hellström, M., 2024. To shred or to disassemble—a techno-economic assessment of automated disassembly vs. shredding in lithium-ion battery module recycling. *Resour. Conserv. Recycl.* 203, 107430.
- Coles, A., Coles, A., Fox, M., Long, D., 2010. Forward-chaining partial-order planning. In: *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 20, pp. 42–49.
- Colledanchise, M., Murray, R.M., Ögren, P., 2017. Synthesis of correct-by-construction behavior trees. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, IEEE*, pp. 6039–6046. <http://dx.doi.org/10.1109/iro.2017.8206502>.
- Colledanchise, M., Parasuraman, R., Ögren, P., 2018. Learning of behavior trees for autonomous agents. *IEEE Trans. Games* 11 (2), 183–189. <http://dx.doi.org/10.1109/tg.2018.2816806>.
- Constantinescu, C., Rus, R., Rusu, C.-A., Popescu, D., 2019. Digital twins of exoskeleton-centered workplaces: Challenges and development methodology. *Procedia Manuf.* 39, 58–65.
- De Graaf, M.M., Dragan, A., Malle, B.F., Ziemke, T., 2021. Introduction to the special issue on explainable robotic systems.
- Destouet, C., Tlahig, H., Bettayeb, B., Mazari, B., 2023. Flexible job shop scheduling problem under industry 5.0: A survey on human reintegration, environmental consideration and resilience improvement. *J. Manuf. Syst.* 67, 155–173.
- Dimitropoulos, N., Togiatis, T., Zacharakis, N., Michalos, G., Makris, S., 2021. Seamless human–robot collaborative assembly using artificial intelligence and wearable devices. *Appl. Sci.* 11 (12), 5699.
- Dworzanski, R., Hlavacs, H., 2024. Shaping ai behavior: A q-learning driven approach to automatic behavior tree creation. In: *2024 IEEE International Conference on Artificial Intelligence and Extended and Virtual Reality. AIXVR, IEEE*, pp. 241–245. <http://dx.doi.org/10.1109/aixvr59861.2024.00039>.
- E. Commission, D.-G. for Research, Innovation, Breque, M., Nul, L. De, Petridis, A., 2021. *Industry 5.0 – Towards a Sustainable, Human-Centric and Resilient European Industry*. Publications Office of the European Union.
- Estgren, M., Jansson, E.S., 2017. *Behaviour Tree Evolution By Genetic Programming*. Linköping University, Linköping, Sweden.
- Eurostat, 2024. Wages and labour costs – statistics explained. [https://ec.europa.eu/eurostat/statistics-explained/index.php?oldid=667135&title=Wages\\_and\\_labour\\_costs](https://ec.europa.eu/eurostat/statistics-explained/index.php?oldid=667135&title=Wages_and_labour_costs).
- Falck, A.-C., Rosenqvist, M., 2014. A model for calculation of the costs of poor assembly ergonomics (part 1). *Int. J. Ind. Ergon.* 44 (1), 140–147.
- Francis, A., 2017. Overcoming pitfalls in behavior tree design. In: *Game AI Pro 3. AK Peters/CRC Press*, pp. 115–125.
- French, K., Wu, S., Pan, T., Zhou, Z., Jenkins, O.C., 2019. Learning behavior trees from demonstration. In: *2019 International Conference on Robotics and Automation. ICRA, IEEE*, pp. 7791–7797. <http://dx.doi.org/10.1109/icra.2019.8794104>.
- George, A.S., George, A., 2020. *Industrial revolution 5.0: the transformation of the modern manufacturing process to enable man and machine to work hand in hand*. J. Seybold Rep. ISSN NO 1533 9211.
- Ghorbani, E., Keivanpour, S., Sekkay, F., Imbeau, D., 2023. *Ergonomic Assembly Line Balancing Problems Evolution and Future Trends with Insights Into Industry 5.0 Paradigm*. Bureau de Montreal, Université de Montreal.
- Greco, A., Caterino, M., Fera, M., Gerbino, S., 2020. Digital twin for monitoring ergonomics during manufacturing production. *Appl. Sci.* 10 (21), 7758.
- Grieves, M., 2014. Digital twin: manufacturing excellence through virtual factory replication. *White Pap.* 1 (2014), 1–7.
- Gurobi Optimization, LLC, 2023. *Gurobi optimizer reference manual*. URL <https://www.gurobi.com>.
- Havard, V., Jeanne, B., Lacomblez, M., Baudry, D., 2019. Digital twin and virtual reality: a co-simulation environment for design and assessment of industrial workstations. *Prod. Manuf. Res.* 7 (1), 472–489.
- Hémono, P., Nait Chabane, A., Sahnoun, M., 2025. Multi objective optimization of human–robot collaboration: A case study in aerospace assembly line. *Comput. Oper. Res.* 174, 106874.
- Hémono, P., Sahnoun, M., Nait Chabane, A., 2023. Optimizing resource allocation in the flexible job shop problem: Assessing the impact of rest breaks on task strenuousness reduction. In: *2023 International Conference on Decision Aid Sciences and Applications. DASA, IEEE*, pp. 320–325.
- Hendrick, H.W., 2008. Applying ergonomics to systems: Some documented lessons learned. *Appl. Ergon.* 39 (4), 418–426.
- Iovino, M., Scukins, E., Styruud, J., Ögren, P., Smith, C., 2022. A survey of behavior trees in robotics and ai. *Robot. Auton. Syst.* 154, 104096.
- Iovino, M., Smith, C., 2023. Behavior trees for robust task level control in robotic applications. <http://dx.doi.org/10.48550/arXiv.2301.06434>, arXiv preprint arXiv: 2301.06434.
- Iovino, M., Styruud, J., Falco, P., Smith, C., 2021. Learning behavior trees with genetic programming in unpredictable environments. In: *2021 IEEE International Conference on Robotics and Automation. ICRA, IEEE*, pp. 4591–4597. <http://dx.doi.org/10.1109/icra48506.2021.9562088>.
- Ippolito, D., Constantinescu, C., Rusu, C.A., 2020. Enhancement of human-centered workplace design and optimization with exoskeleton technology. *Procedia CIRP* 91, 243–248.
- Karhu, O., Kansu, P., Kuorinka, I., 1977. Correcting working postures in industry: A practical method for analysis. *Appl. Ergon.* 8 (4), 199–201. [http://dx.doi.org/10.1016/0003-6870\(77\)90164-8](http://dx.doi.org/10.1016/0003-6870(77)90164-8), URL <https://www.sciencedirect.com/science/article/pii/0003687077901648>.
- Karuppiah, K., Sankaranarayanan, B., Ali, S.M., Bhalaji, R., 2023. Decision modeling of the challenges to human–robot collaboration in industrial environment: a real world example of an emerging economy. *Flex. Serv. Manuf. J.* 35 (4), 1007–1037.
- Kheirabadi, M., Keivanpour, S., Chinniah, Y.A., Frayret, J.-M., 2023. Human–robot collaboration in assembly line balancing problems: Review and research gaps. *Comput. Ind. Eng.* 186, 109737.
- Klößner, A., 2013. *Behavior Trees for Uav Mission Management, INFORMATIK 2013: Informatik Angepasst an Mensch. Organisation und Umwelt*, pp. 57–68.
- Kritzinger, W., Karner, M., Traar, G., Henjes, J., Sihm, W., 2018. Digital twin in manufacturing: A categorical literature review and classification. *Ifac-PapersOnline* 51 (11), 1016–1022.
- Li, J., Barwood, M., Rahimifard, S., 2018. Robotic disassembly for increased recovery of strategically important materials from electrical vehicles. *Robot. Comput.-Integr. Manuf.* 50, 203–212.
- Lim, C.-U., Baumgarten, R., Colton, S., 2010. Evolving behaviour trees for the commercial game defcon. In: *Applications of Evolutionary Computation: EvoApplications 2010: EvoCOMPLEX, EvoGAMES, EvoIASP, EvoINTELLIGENCE, EvoNUM, and EvoSTOC, Istanbul, Turkey, April (2010) 7–9, Proceedings, Part I*. Springer, Springer Berlin Heidelberg, pp. 100–110. [http://dx.doi.org/10.1007/978-3-642-12239-2\\_11](http://dx.doi.org/10.1007/978-3-642-12239-2_11).
- Lou, P., Liu, Q., Zhou, Z., Wang, H., Sun, S.X., 2012. Multi-agent-based proactive-reactive scheduling for a job shop. *Int. J. Adv. Manuf. Technol.* 59, 311–324.
- Lou, S., Zhang, Y., Tan, R., Lv, C., 2024. A human-cyber–physical system enabled sequential disassembly planning approach for a human–robot collaboration cell in industry 5.0. *Robot. Comput.-Integr. Manuf.* 87, 102706.
- Malik, A.A., et al., 2024. The economic impact of digital twin technology on manufacturing systems: A practical evaluation framework. In: *Proceedings of the Winter Simulation Conference. WSC, Orlando, FL*.

- Marketsandmarkets.com, 2020. Digital twin market by source, type, application & geography | covid-19 impact analysis | marketsandmarkets™. <https://www.marketsandmarkets.com/Market-Reports/digital-twin-market-225269522.html>. (Accessed 01 June 2025).
- Martín, F., Clavero, J.G., Matellán, V., Rodríguez, F.J., 2021. Plansys2: A planning system framework for ros2. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, IEEE, pp. 9742–9749. <http://dx.doi.org/10.1109/iro51168.2021.9636544>.
- Nait Chabane, A., Hémono, P., Sahnoun, M., Benali, A., 2023. Fatigue evaluation and scheduling for manual tasks: A break planning approach. CIE 2023, In: 50th International Conference on Computers and Industrial Engineering, vol. 1.
- Nkisi-Orji, I., Paliawadana, C., Wiratunga, N., Wijekoon, A., Corsar, D., 2023. Failure-driven transformational case reuse of explanation strategies in cloudcbr. In: International Conference on Case-Based Reasoning. Springer, pp. 279–293. [http://dx.doi.org/10.1007/978-3-031-40177-0\\_18](http://dx.doi.org/10.1007/978-3-031-40177-0_18).
- Ogunseju, O.R., Olayiwola, J., Akanmu, A.A., Nnaji, C., 2021. Digital twin-driven framework for improving self-management of ergonomic risks. Smart Sustain. Built Environ. 10 (3), 403–419.
- Ouelhadj, D., Petrovic, S., 2009. A survey of dynamic scheduling in manufacturing systems. J. Sched. 12, 417–431.
- Özgülven, C., Özbakır, L., Yavuz, Y., 2010. Mathematical models for job-shop scheduling problems with routing and process plan flexibility. Appl. Math. Model. 34 (6), 1539–1548. <http://dx.doi.org/10.1016/j.apm.2009.09.002>.
- Perez, D., Nicolau, M., O'Neill, M., Brabazon, A., 2011. Evolving behaviour trees for the mario ai competition using grammatical evolution. In: European Conference on the Applications of Evolutionary Computation. Springer, Springer Berlin Heidelberg, pp. 123–132. [http://dx.doi.org/10.1007/978-3-642-20525-5\\_13](http://dx.doi.org/10.1007/978-3-642-20525-5_13).
- Peruzzini, M., Grandi, F., Pellicciari, M., 2020. Exploring the potential of operator 4.0 interface and monitoring. Comput. Ind. Eng. 139, 105600.
- Pi, R., Cieślak, P., Esteba, J., Palomeras, N., Ridao, P., 2023. Compliant manipulation with quasi-rigid docking for underwater structure inspection. IEEE Access 11, 128957–128969. <http://dx.doi.org/10.1109/access.2023.3332486>.
- Premevida, C., Ambrus, R., Marton, Z.-C., 2018. Intelligent robotic perception systems. Appl. Mob. Robot. 111–127.
- Pusch, A., Noël, F., 2019. Augmented reality for operator training on industrial workplaces—comparing the microsoft hololens vs. small and big screen tactile devices. In: Product Lifecycle Management in the Digital Twin Era: 16th IFIP WG 5.1 International Conference, PLM 2019, Moscow, Russia, July (2019) 8–12, Revised Selected Papers 16. Springer, pp. 3–13.
- R. d. P. Pereira, Engel, P.M., 2015. A framework for constrained and adaptive behavior-based agents. <http://dx.doi.org/10.48550/arXiv.1506.02312>, arXiv preprint arXiv:1506.02312.
- Rega, A., Di Marino, C., Pasquariello, A., Vitolo, F., Patalano, S., Zanella, A., Lanzotti, A., 2021. Collaborative workplace design: A knowledge-based approach to promote human–robot collaboration and multi-objective layout optimization. Appl. Sci. 11 (24), 12147.
- Renke, L., Piplani, R., Toro, C., 2021. A review of dynamic scheduling: context, techniques and prospects, implementing industry 4.0: The model factory as the key enabler for the future of manufacturing. pp. 229–258.
- Rosen, R., Von Wichert, G., Lo, G., Bettenhausen, K.D., 2015. About the importance of autonomy and digital twins for the future of manufacturing. Ifac-Papersonline 48 (3), 567–572.
- Roy, R., Stark, R., Tracht, K., Takata, S., Mori, M., 2016. Continuous maintenance and the future—foundations and technological challenges. Cirp Ann. 65 (2), 667–688.
- Ruiz-Celada, O., Verma, P., Diab, M., Rosell, J., 2022. Automating adaptive execution behaviors for robot manipulation. IEEE Access 10, 123489–123497.
- Sahnoun, M., Baudry, D., Mustafee, N., Louis, A., Smart, P.A., Godsiff, P., Mazari, B., 2019. Modelling and simulation of operation and maintenance strategy for offshore wind farms based on multi-agent system. J. Intell. Manuf. 30 (8), 2981–2997.
- Sanogo, K., Benhafssa, A.M., Sahnoun, M., 2025. A game theory approach for optimizing job shop scheduling problems with transportation in common shared human–robot environments. Comput. Ind. Eng.
- Scheper, K.Y., Tijmons, S., de Visser, C.C., de Croon, G.C., 2016. Behavior trees for evolutionary robotics. Artif. Life 22 (1), 23–48. [http://dx.doi.org/10.1162/artl\\_a\\_00192](http://dx.doi.org/10.1162/artl_a_00192).
- Scherf, L., Schmidt, A., Pal, S., Koert, D., 2023. Interactively learning behavior trees from imperfect human demonstrations. Front. Robot. AI 10, <http://dx.doi.org/10.3389/frobt.2023.1152595>.
- Segura-Muros, J.Á., Fernández-Olivares, J., 2017. Integration of an automated hierarchical task planner in ros using behaviour trees. In: 2017 6th International Conference on Space Mission Challenges for Information Technology. SMC-IT, IEEE, pp. 20–25. <http://dx.doi.org/10.1109/smc-it.2017.11>.
- Sgarbossa, F., Grosse, E.H., Neumann, W.P., Battini, D., Glock, C.H., 2020. Human factors in production and logistics systems of the future. Annu. Rev. Control. 49, 295–305.
- Sharma, A., Kosasih, E., Zhang, J., Brintrup, A., Calinescu, A., 2022. Digital twins: State of the art theory and practice, challenges, and open research questions. J. Ind. Inf. Integr. 30, 100383.
- Tao, F., Cheng, J., Qi, Q., Zhang, M., Zhang, H., Sui, F., 2018a. Digital twin-driven product design, manufacturing and service with big data. Int. J. Adv. Manuf. Technol. 94, 3563–3576.
- Tao, F., Zhang, H., Liu, A., Nee, A.Y., 2018b. Digital twin in industry: State-of-the-art. IEEE Trans. Ind. Inform. 15 (4), 2405–2415.
- Tao, J., Zhu, X., Ma, L., Wu, M., Bao, W., Wang, J., 2020. Benign: An automatic optimization framework for the logic of swarm behaviors. In: 2020 IEEE International Conference on Systems, Man, and Cybernetics. SMC, IEEE, pp. 2999–3005. <http://dx.doi.org/10.1109/smc42975.2020.9283409>.
- Tarallo, A., Mozzillo, R., Di Gironimo, G., De Amicis, R., 2018. A cyber–physical system for production monitoring of manual manufacturing processes. Int. J. Interact. Des. Manuf. (IJIDeM) 12, 1235–1241.
- Thomas, D., 2024. Economics of Digital Twins: Costs, Benefits, and Economic Decision Making. Tech. Rep. AMS 100-61, National Institute of Standards and Technology, <http://dx.doi.org/10.6028/NIST.AMS.100-61>.
- Uhlemann, T.H.-J., Lehmann, C., Steinhilper, R., 2017. The digital twin: Realizing the cyber–physical production system for industry 4.0. Procedia Cirp 61, 335–340.
- Wu, S.D., Byeon, E.-S., Storer, R.H., 1999. A graph-theoretic decomposition of the job shop scheduling problem to achieve scheduling robustness. Oper. Res. 47 (1), 113–124.
- Zafar, M.H., Langås, E.F., Sanfilippo, F., 2024. Exploring the synergies between collaborative robotics, digital twins, augmentation, and industry 5.0 for smart manufacturing: A state-of-the-art review. Robot. Comput.-Integr. Manuf. 89, 102769.