sustainability

Article

Autonomous Agent-Based Adaptation of Energy-Optimized Production Schedules Using Extensive-Form Games

William Motsch [1,*], Achim Wagner [2] and Martin Ruskowski [2]

[1] Technologie-Initiative SmartFactory KL e.V., Trippstadter Str. 122, 67663 Kaiserslautern, Germany
[2] German Research Center for Artificial Intelligence (DFKI), Trippstadter Str. 122, 67663 Kaiserslautern, Germany; achim.wagner@dfki.de (A.W.); martin.ruskowski@dfki.de (M.R.)
* Correspondence: william.motsch@smartfactory.de

Abstract: Modular cyber-physical production systems are an important paradigm of Industry 4.0 to react flexibly to changes. The flexibility of those systems is further increased with skill-based engineering and can be used to adapt to customer requirements or to adapt manufacturing to disturbances in supply chains. Further potential for application of these systems can be found in the topic of electrical energy supply, which is also characterized by fluctuations. The relevance of energy-optimized production schedules for manufacturing systems in general becomes more important with the increased use of renewable energies. Nevertheless, it is often difficult to adapt when short-term energy price updates or unforeseen events occur. To address these challenges with an autonomous approach, this contribution focuses on extensive-form games to adapt energy-optimized production schedules in an agent-based manner. The paper presents agent-based modeling to transform and monitor energy-optimized production schedules into game trees to respond to changing energy prices and disturbances in production. The game is setup with a scheduler agent and energy agents who are considered players. The implementation of the mechanism is presented in two use cases, realizing decision making for an energy price update in a simulation example and for unforeseen events in a real-world demonstrator.

Keywords: production planning; multi-agent systems; energy-aware decision making; extensive-form games; modular skill-based production

Citation: Motsch, W.; Wagner, A.; Ruskowski, M. Autonomous Agent-Based Adaptation of Energy-Optimized Production Schedules Using Extensive-Form Games. *Sustainability* **2024**, *16*, 3612. https://doi.org/10.3390/su16093612

Academic Editors: Silviu Răileanu, Theodor Borangiu and Damien Trentesaux

Received: 29 February 2024
Revised: 8 April 2024
Accepted: 18 April 2024
Published: 25 April 2024

1. Introduction

Industry 4.0 enables the consideration of customer-specific criteria during manufacturing, to produce profitably even with smaller batch sizes, and additionally offers the potential to include topics of resource efficiency, such as the integration of energy consumption into the planning and operation of production facilities [1]. The sustainability aspects of energy management and the use of energy data in production planning are as well of key importance in the field of smart factories [2]. These aspects can be placed into the potential that comes with the concept of demand side management (DSM) of industrial energy consumption with a dynamic price-based use of renewable energy [3].

Industry 4.0 can currently be viewed as part of a development period, in which an autonomous production control is targeted [4]. Autonomous systems can be used and interlinked in smart factories to meet the requirements of adaptable and flexible production, based on the availability of cyber-physical systems (CPS) with their further development, which leads to the implementation of those autonomous systems [5]. Planning is especially important for autonomous systems, whose characteristics include the ability to make decisions and to adapt to available resources [6]. The aspects of energy-efficient planning thus must be placed in the context of these developments.

Modular production systems are controllable more autonomously by using multi-agent systems (MAS), enabling software agents to access the encapsulated capabilities of several production modules [4]. Besides the focus on controlling production modules in distributed production environments, the relevance of integrating humans and machine learning algorithms is important for current MAS developments, as presented in the MAS4AI framework [7]. The relevance of machine learning agents in such frameworks as MAS4AI can be found in the application of production scheduling in combination with Industry 4.0 technologies [8]. A special agent type, which belongs to production modules, are energy agents, which are developed and realized within MAS4AI for the creation of energy load profiles in modular production systems [9]. For the next steps of development and research, interaction and communication with planning agents and scheduling agents were mentioned in [9].

Contributions to the adaptation of energy loads from energy consumers are well known in the domain of game theory, where for DSM, a formalization as an energy consumption game can be applied, as presented in [10,11]. The application of game theory for energy-aware scheduling can also be found in the domain of cloud computing, in which the game could be designed as a Stackelberg game, considering a workload scheduler and energy efficiency agents as its own players, to adapt the scheduler decisions with energy-optimization techniques [12]. With regard to the implementation and optimization of modular production systems, the application of game theory with players as an analogy to agents is feasible, whereby, the open question for further research is how energy agents for Stackelberg games could be designed and developed [13]. A multi-criteria flexible job-shop scheduling (FJSS) with the application of game theory can consider energy consumption, energy efficiency, and unforeseen events, in a system that is based on a pre-optimization layer and in which a game tree representation is used to react to changes [14].

A structured literature review about energy-oriented decision support in production environments pointed out that there is potential for future research regarding the topic of rescheduling, especially in the context of flexible energy prices [15]. In relation to the introduced topics of game theory and agent-inspired contributions, game trees can be formalized and used to adapt already calculated and pre-optimized production schedules, to react with the players of the game tree to unforeseen events under consideration of energy efficiency [16]. The role of decision trees in combination with machine learning and responsible AI and their purpose for explainability to humans are generally of importance [17]. The agent-based transformation and execution of schedules, based on a game tree formalism, enables the application of established tree-search mechanisms. In game tree search within game theory, parallelization and pruning mechanisms can be applied to achieve faster solutions [18]. Furthermore, Monte Carlo tree search (MCTS) can be used for industrial scheduling in combination with machine learning [19].

The current contribution is mainly based on the preliminary research from [9,16] and continues with the opportunities for future research on the design of mechanisms between scheduler agents and energy agents. The topics of realizing an energy-efficient production in modular production systems by using game trees, especially for rescheduling purposes, are examined. The agent-based adaptation of already energy-optimized production schedules with game tree mechanisms is focused on reacting on unforeseen events, such as changing energy prices. The main aim of the current contribution is to show that a human-understandable autonomous decision-based rescheduling of production schedules is possible, with the combined conceptual approaches from agents of industrial-focused MAS and applied extensive-form games of game theory. The concept focuses on the general elements that are necessary to implement the system and presents with rescheduling use cases how a schedule adaptation can be modeled and realized.

The paper structure is as follows: In Section 2, a literature review is presented, based on the introduced literature topics, to highlight the current state of the art and research as related work. In Section 3, the specification of the overall system is shown, including the setup of the MAS framework, using an instantiation of the MAS4AI framework, and the design for both focused agent types, the scheduling agent, and the energy agent. Section 4 shows the concept of the agent-based adaptation of energy-optimized production schedules and the application of schedule examples. In Section 5, results are presented for rescheduling use cases of a simulated energy price adaptation and for skill execution in a real-world demonstrator. In Section 6, the results of the contribution are discussed. Section 7 summarizes the contribution with a conclusion and opportunities for future research.

2. Literature Review

The literature that is applied in this contribution is based on the topics of modular production systems within Industry 4.0, the encapsulation of functionalities of such systems with a skill-based approach, the modeling of energy-efficient production and its consideration in planning and scheduling, the developments of distributed agent-control in manufacturing environments, and lastly, the fundamentals of extensive-form games in game theory. The theoretical basics and related works are shown for these topics, to set the literature-based scope for the current contribution.

2.1. Skill-Based Cyber-Physical Production Systems

CPS are a main element towards the goal of a more autonomous production [5]. A CPS can provide an autonomous control and is important for the design of smart factories [20]. The role of CPS for manufacturing within Industry 4.0 is discussed broadly and consists of various challenges, approaches, and techniques used, also regarding the domain of MAS [21]. A modular and flexible production environment is a cyber-physical production system (CPPS) and can be built with a combination of various cyber-physical production modules (CPPM) [22]. A CPPM can be defined as a module that contains three logically aggregated entities [23]:

- An equipment,
- a controller or a computing platform, and
- a cyber representation of the whole.

The relevance of such a modularized production can furthermore be seen in the application of the principle of subsidiarity, following a decentralized structure with autonomous acting production units [24]. CPPMs can be built from several CPS and are thus an elemental concept of modular production environments, providing standardized interfaces for different functionalities with their own services [22]. The usage of a standardized interface design can thus support interacting with vendor-independent production modules from the perspective of a human worker [25].

The cyber representation mentioned in [23] contains the interface and the algorithms for the module interaction with other modules without the requirement of reprogramming, and, furthermore, the implementation of a hardware abstraction layer for decoupling the interaction and execution logic from the equipment. An important concept to provide production functionalities with such interfaces is described within the capability, skill, and service model (CSS model), with the aim of providing a higher level of adaptability and flexibility [26]. The definition of a skill is stated to be an asset-dependent implementation of an asset-independent production capability [26,27]. The mechanism of utilizing production capabilities to match product requirements with suitable production resources can be implemented with the elements of the CSS model [26]. It is furthermore a main element to realize a shared production paradigm and to provide and use decentralized production services on demand among a network of trustworthy partners [24].

In the scope of the current contribution, skills are considered as an abstraction layer that enables a uniform interface concept for interaction and communication with production systems. It follows from the design of [4] that several CPPMs can provide their skills, such as assembly or quality check, to produce a product, which are useable by distributed organized control software realized with a MAS. The formalized concept of the current contribution on how such production skills can be applied in a MAS is based on the framework approach from [7], which has been extended for energy-related functionalities in [9].

2.2. Energy-Efficient Production Optimization within Industry 4.0

From an ecological perspective, resource efficiency and energy efficiency have been mentioned as key objectives since the beginning of Industry 4.0 [28]. Energy in Industry 4.0 plays an important role in the topic of resource efficiency, considering energy consumption explicitly for planning purposes [1]. The topic of energy management and the application of energy data for production planning are important for smart factories [2]. The domains of energy-efficient scheduling in intelligent production systems [29] and manufacturing companies [30] are considered in literature reviews. To integrate and apply energy-efficient planning and scheduling in CPPS, a consideration of several aspects is necessary. Energy flexibility of manufacturing systems is important for the planning and shifting of production orders, so that energy consumption can be considered in relation to processing time [31]. Different components are required to realize the mechanism of a DSM for electrical energy in a modular organized production, in which energy data must be measured, preprocessed, and stored related to information about products, processes, and resources [32]. Energy consumption data must be accessible for a planning and scheduling component, in combination with information about energy prices in the time horizon of the schedule. The application of DSM for energy consumption in an industrial context can thus adapt renewable energy through dynamic prices [3].

The topic of energy-efficient scheduling is of interest in literature and research. Energy consumption measurements provide the basis for the optimization of energy consumption during a job-shop manufacturing process and can be used to realize a decision-making algorithm that is based on a decentralized system [33]. The optimization of production systems for optimal load control and scheduling can be performed using distributed mixed-integer linear programming (MILP), in which energy-flexible production units can choose between several intensities for processing [34]. The improvement of energy consumption in manufacturing systems can be considered through the scheduling of a machine startup and shutdown, in which the tradeoff between productivity and energy efficiency must be discussed and in which the energy-efficient production problem can be formalized as a constrained optimization problem [35]. A reactive scheduling approach for flexible manufacturing systems, which integrates the energy consumption of these systems, can also use the mechanism of potential fields, so that resources can detect the intentions of products and decide on a standby mode to save energy [36].

From a more decision-based point of view, a structured literature review shows that in the field of energy-related decision support, the topic of rescheduling for unforeseen events is not well considered in the literature, where also dynamic energy prices are in general as considerable as such events [15]. Despite the number of works in the field of energy consumption in the planning domain of manufacturing, there are still open questions that must be combined with several modeling approaches for different manufacturing systems, especially with decision-support models that take care of mid-term and short-term production planning [37]. A decision-based application with extensive-form games for scheduling in manufacturing under energetical viewpoints, which is used to reduce the complexity of the multi-objective flexible job-shop scheduling problem (FJSSP) and to improve the solving efficiency, is based on a pre-optimization layer [14]. A formalized utilization of extensive-form games with game trees for modular skill-based production environments is possible, in which a schedule can be adapted by production units, which are able to adapt to the decisions from a pre-optimized scheduling [16]. This concept focuses on local decisions on how the production could be performed best in an energetic way, for example, by choosing intensities for active timeframes and shutdown policies for passive timeframes.

Modeling energy characteristics for planning and scheduling applications in modular production environments is an important topic, where the flexibility of production units is an essential aspect. The usage of decision-based solutions proves a high potential to react to unforeseen events, so that decision support and concepts from game theory with games in extensive form and game trees have potential for rescheduling purposes.

2.3. Multi-Agent Systems and Holonic Manufacturing Systems for Sustainable Production

In recent years, various definitions of agents and research have been made in the field of MAS. An agent can be defined as a computer system in a certain environment that can act autonomously to achieve its goal [38]. Furthermore, an agent can be described as an autonomous component, representing physical or logical objects in a system, which can act with the intention to reach its goals, and can interact with other agents, when their own knowledge and skills are not sufficient to reach the objectives [39]. An agent in relation to CPS can be described as anything that perceives its environment with sensors and that uses actuators to influence this environment [40]. A more generalized definition considers the fundamental abilities and features of agents, so that an agent is defined as an entity that is placed in an environment and perceives various parameters to make decisions based on its goal, and to perform required actions on the environment based on this decision [41]. A MAS is described as a system consisting of several agents that interact with each other, typically by exchanging messages through a computer network infrastructure [38]. Agents in a MAS depend on communication, collaboration, negotiation, and delegation of responsibility to be successful [42].

A further paradigm, which must be considered in this context, is the concept of holonic manufacturing systems (HMS). The manufacturing paradigm of HMS has been developed with the goal of improving the adaptability of manufacturing systems [43]. HMS are fundamentally described in the PROSA reference architecture, which covers hierarchical and heterarchical control approaches for production systems [44]. A holon itself can be characterized as an autonomous, intelligent, and cooperative element of a manufacturing system [45]. As described in [43], a holon is stated as a special type of agent, and the technology used for holonic systems is mostly the MAS. A holonic architecture for agile and adaptive manufacturing control is provided with ADACOR, in which the main elements are modeled as holons [46]. There are contributions and challenges to holonic control architectures in the context of Industry 4.0, considering autonomous and decentralized decision support as key enablers for manufacturing systems in Industry 4.0 [47]. The concept of a holonic structured system that is realized by MAS for production applications is also used in the MAS4AI framework, under consideration of technologies from Industry 4.0 and focusing on the integration of humans, CPPMs, and algorithms with a skill-based approach [7]. Literature surveys focus on the topic of MAS with CPS in industrial applications, also considering MAS architectures for the production domain [48]. A CPPS architecture based on multi-agent design patterns with the requirements of Industry 4.0 and several MAS architectures is proposed in [49].

With the importance of energy-efficient production scheduling, further possibilities arise to integrate sustainable viewpoints into the design of holonic structured MAS and to combine them with the technologies from Industry 4.0. With ADMARMS, a design method is described to create MAS for production systems [50]. Based on the increasing need for sustainable developments, methods are presented to design a sustainable intelligent manufacturing control system with holonic multi-agents [51]. The realization of a shared production paradigm also considers topics of sustainability and autonomous control of skill-based production environments with agents [24].

The topic of specialized energy agents for flexible and reconfigurable CPS is of importance within the context of Industry 4.0 [52]. A prototypical development of energy agents in combination with resource agents for modular skill-based CPPS has been validated using the MAS4AI framework [53]. The concept of the energy agent was subsequently adopted and further developed, realizing agent-based energy data acquisition for the creation of energy load profiles on a detailed level of production skills in modular environments, in which especially as a future research opportunity, the combination with scheduler agents was mentioned [9]. The agent-based consideration thus must focus on the energy data of modularized production environments to enable further applications, such as rescheduling based on energy aspects, for example, changes in dynamic energy prices.

2.4. Extensive-Form Games in Game Theory

The application of game theory for scheduling and the adaptation of such schedules is of importance in distributed environments and provides the potential to take energy-related objectives into account. Concepts of distributed optimization and game-theoretic application can also be found in the domain of intelligent energy supply networks such as smart grids. Energy consumption games can be formalized and simulated to realize an autonomous system for DSM, in which the players participate as users in dynamic price tariffs for electrical energy by using the strategy of their daily schedules for their household [10]. DSM can be formalized as an energy consumption game, in which energy suppliers adapt a day-ahead price strategy from a smart grid and then publish price information to several customers with the purpose of planning household equipment's to minimize energy costs [11].

Since the concepts of agents from the domain of MAS and players from the domain of game theory are similar in several aspects, there is research potential for the application of Stackelberg games, in which energy agents can react to schedule-related decisions with their own decisions from an energy-related perspective [13]. Stackelberg games, in which one player moves first and then the other players move afterwards, by choosing the best answer to the previous decision, can also be found in the domain of cloud computing, where one player acts as a workload scheduler and the other players, with their computing units, try to adapt to this decision with the best energy-optimization decision, realizing a shutdown policy [12]. Since this sequential characteristic of Stackelberg games can be formalized by using game trees, the topic of game trees and the solution concepts of those games can be elaborated on in general for their application towards scheduling adaptations.

Of major importance in game theory is the Nash equilibrium, which was defined and proven by John Forbes Nash Jr., so that in non-cooperative games there is a strategy combination that is chosen by the players and from which there is no improvement for any player to be the only one to deviate from it [54]. Solution concepts for cooperative and non-cooperative games exist, in which partially perfect and sequential equilibria are considered in extensive and dynamic games [55]. Furthermore, finite games with complete information can contain subgames, in which subgame-related Nash equilibria also exist [56]. With a game tree in extensive form, dynamic decision situations can be analyzed, in which players make their actions dependent on information that they only received during their interactions, although the overall game can be broken down into individual subgames for this purpose [55]. The extensive form is a generalization of a decision tree for several players [57]. The application of game theory for a multi-criteria FJSS under energetical aspects and unforeseen events by using game trees and subgames is feasible for decision making in manufacturing [14]. The usage of an extensive-form game for an energy-aware adaptation of production schedules, similar to the general Stackelberg game mechanism from [12], is shown for a modular production setup in [16], where CPPMs can choose energy-related capabilities to react to decisions of the scheduler.

The application of tree-search mechanisms to solve FJSSPs in manufacturing systems in general is of importance in recent literature, for which MCTS algorithms can be applied, as presented in [58]. Algorithm developments are proposed for industrial scheduling with MCTS in combination with machine learning [19]. MCTS methods are also considered in literature surveys [59]. The application of alpha-beta algorithms for game tree search can also be considered from early contributions, featuring the general parallelization perspective of such algorithms as described in [60]. In general, strategies for distributed game tree search must be considered, for example, as shown in [61]. The potential of MCTS can be examined in recent implementations, for example, as shown in the implementation of [62], which can also be used for simulation optimizations [63].

The realization of an autonomous agent-based extensive-form game between scheduler agents and energy agents following similar principles to Stackelberg games is considered in the current contribution.

2.5. Literature Summary and Research Potential

The literature review focuses on the literature topics of skill-based CPPS in Industry 4.0 and the importance of energy-efficient production optimization, especially considering the topics of scheduling and rescheduling. Related to energy-aware scheduling, the importance of decision support to react to changing energy prices is of interest in upcoming research. Since centralized production scheduling models must consider many aspects, there is a lot of complexity that must be handled, which could also influence computation times. Applied to a modular and flexible structure of CPPS, it is in general possible to enable a distributed optimization and computation, since each CPPM of such a system encapsulates its functionality and has its own computational abilities. Nevertheless, centralized optimization models are of importance and are often able to find an optimal or near-optimal solution for a production schedule in a suitable computation time, unless changes must be considered. Solutions for similar problems can be found in the domain of heterarchical MAS, since the initial planning of such systems tries to find the best solution for the system, which is provided by agents of a higher hierarchical level. In cases of disturbances, the agents at a lower hierarchical level try to solve the problems locally and communicate with other agents. Since decision making in such heterarchical systems could as well be difficult to follow, a mechanism is required to combine autonomous rescheduling with the necessity of providing decisions in a human-readable way.

The game theoretical approach that has been chosen in the scope of the current contribution is utilized to address those aspects and is applicable for energy-aware rescheduling in modular production environments. Additionally, the introduced mechanism of Stackelberg games with usage of energy prices is used in literature for similar problems that a smart grid is confronted with, for example, to implement a mechanism for DSM. The combination of scheduling agents and energy agents in production is suitable for the development of a human-readable and transparent decision-making approach with a heterarchically organized MAS, which tries to reach autonomous adaptation to changing energy prices and unforeseen events. The mainly considered contributions from the literature review and their aspects for future research potential are summarized in Table 1.

Table 1. Literature overview of the mainly considered contributions from the literature review with their future research potential.

Author(s)	Literature Topics	Publication Content and Focus	Future Research Potential
Dafflon et al. (2021) [21]	Industry 4.0, CPS in combination with MAS	Literature review about developments of CPS	Integration of the cyber-component of CPS
Bergweiler et al. (2022) [24]	Industry 4.0, modular distributed manufacturing systems	Vision for production systems of the future	Autonomous sustainable manufacturing using MAS
Diedrich et al. (2022) [26]	Capability, skills and service model	Information model for manufacturing	Standardized skill models
Bänsch et al. (2021) [15]	Energy-aware decision support models in production	Literature review about energy decisions in production	Consideration of energy price changes in decisions
Sidorenko et al. (2023) [7]	MAS for smart manufacturing, AI algorithms	MAS framework for human-centered manufacturing	Planning and optimization
Motsch et al. (2023) [9]	Energy agents, modular distributed manufacturing systems, skill models	Energy agents for load profiling in a modular production	Planning and optimization
Schwung (2021) [13]	Game theory, modular distributed manufacturing systems	Application of game theory in modular production	Development of Stackelberg games with energy agents
Zhang et al. (2017) [14]	Game theory, flexible job-shop scheduling, energy optimization	Application of game theory for energy-aware rescheduling	Realization with agents
Motsch et al. (2023) [16]	Energy-aware decision support, game theory, skill models	Energy-related skills, extensive-form games for rescheduling	Realization with agents

3. Modeling and Realization of the MAS-Framework

The concept for the implementation and realization of an agent-based energy-aware production schedule adaptation is an important part of the current contribution. First, in Section 3.1, the general setup of the MAS4AI framework is presented, which is applicable for the agent-based mechanism in a modular and decentralized Industry 4.0 production environment. Furthermore, an overview of the required components is provided. Section 3.2 then focuses on the agent type models for the scheduler agent and the energy agent in a formalized way, introducing how agent behaviors and agent skills can be used to enable them as players in game theory, realizing a scheduling adaptation with extensive-form games.

3.1. Setup and Instantiation of the MAS4AI Framework

The agent-based structure that is specified for the current contribution follows the concept of the MAS4AI framework. The MAS4AI framework provides a generalized concept on how a MAS can be applied for the usage of AI algorithms and human integration in Industry 4.0 production environments and thus considers research in the field of industrial MAS and HMS [7]. The MAS4AI framework can be designed and setup individually, introducing the main concept as an abstracted framework with designated components, which provides the possibility to select and instantiate suitable technological solutions [64]. The MAS4AI framework is specifically instantiated in the scope of the current contribution. The related components are introduced to present a proposal for the instantiation of this system and for the realization of the agent-based energy-aware production schedule adaptation. The applicability of the framework and the research results for modular skill-based production environments were implemented and demonstrated in the SmartFactory-KL testbed environment, considering several agent types that are used to plan and control production [53]. The components of the MAS4AI framework used for the current contribution are visualized in Figure 1.

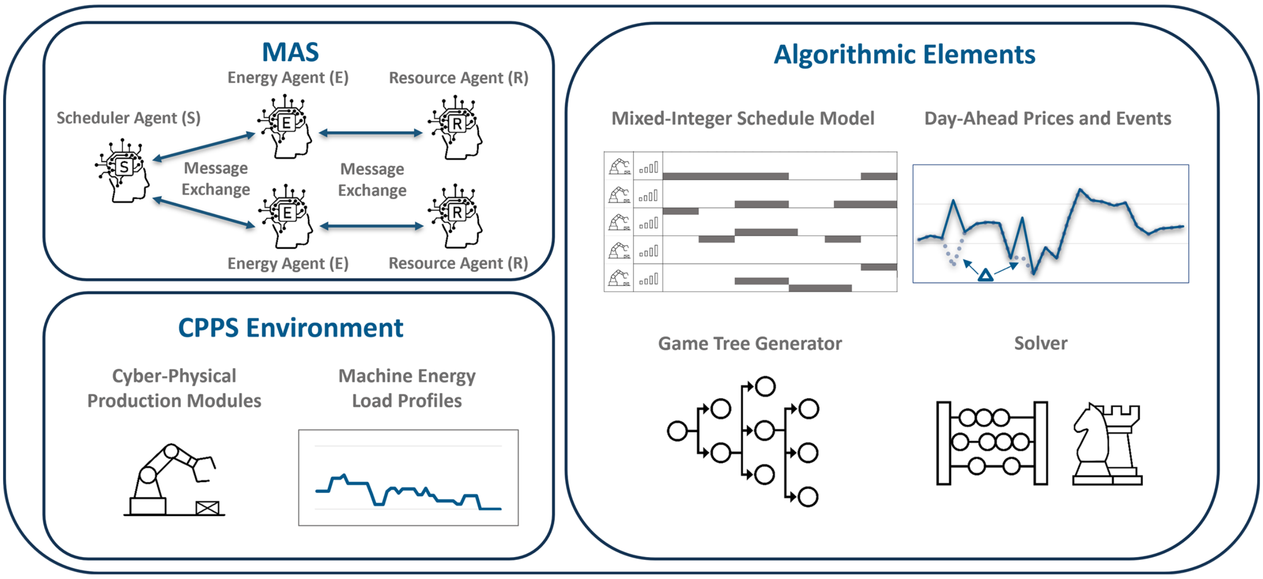


Figure 1. MAS4AI framework instance, based on the proposed framework concept of [7].

The several components of the system were chosen and instantiated in the same way as proposed in [7], as described in [64], and evaluated in [53]. The MAS component was implemented with the open-source Janus framework, which implements the agent-oriented programming language SARL and provides the possibility to develop holonic agents [65]. With the Janus framework, which is used for the development of the MAS for the SmartFactory-KL testbed environment, agent types such as planning agents, scheduling agents, resource agents for manufacturing control, as well as energy agents for energy load profiling, have been developed, implemented, and tested [53]. These agent types are applied and extended for the suggested agent-based energy-aware production schedule adaptation. The implementation of energy agents has been examined more in detail in the contribution of [9], in which the MAS4AI framework was applied, instantiated, and tested with reference to the following components:

- Janus framework implementation as MAS with resource agents, which have the aim to fulfill a production schedule, to control CPPMs, and communicate with energy agents, which can access the related metering infrastructure.
- The usage of the Asset Administration Shell (AAS) as a standardized interface for agents, also considering skill interfaces of modular production units as well as for infrastructure units.
- Data acquisition of production units with specific relation for energy load profiling during production in a methodical way, using and adapting the proposed method from [66] for skill-based manufacturing.

From the perspective of the instantiated components of the MAS4AI framework, the Janus framework is used as MAS for the following agent types: Scheduler, energy, and resource. In Figure 2, the activity view of the scheduler agent is shown.

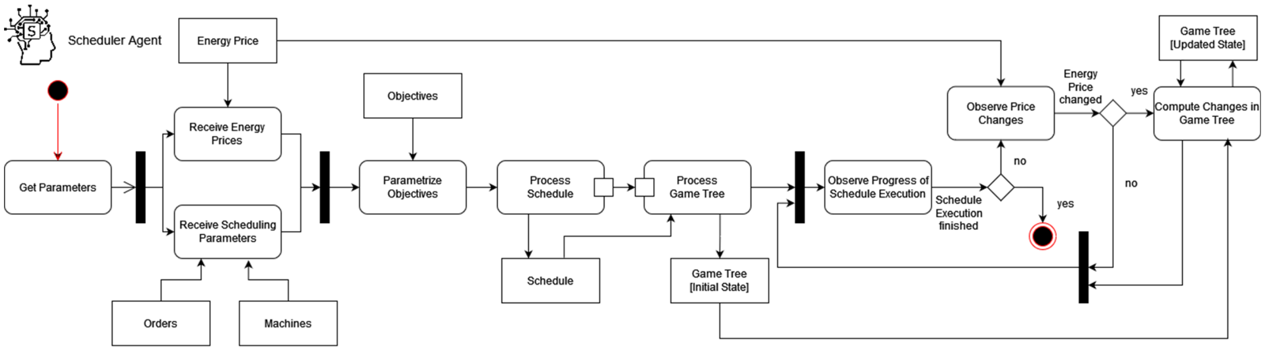


Figure 2. Activity diagram for energy-aware scheduling of the scheduler agent of a CPPS.

The scheduler agent uses, for the initial scheduling of the CPPS, a multi-objective FJSSP-Scheduler as a pre-scheduling software component. This component provides an initial schedule under consideration of the criteria for an overall makespan as well as the related energy consumption with usage of day-ahead energy prices. It is important that this component provide an optimal or near-optimal solution for the scheduling problem. For the instantiated solution, a MILP model is applied. If at least a full schedule is provided as an outcome of this component, several algorithms are useable to solve the scheduling problem, for example, with a distributed MILP as shown in [34]. The outcome of the optimized schedule is then used in the game tree schedule component, in which the schedule from the FJSSP-Scheduler can be transferred in parallel to a game tree formalism, using the game tree generation algorithm as shown in [16]. Another component that is required by the scheduler agent is the interface for the dynamic energy prices, which is required to obtain energy price information for the day-ahead schedule in the FJSSP-Scheduler component and to obtain information for short-term price changes, which result in the application for rescheduling of the previous generated game tree, to change decisions on parts of the schedule locally. The energy price interface component is designed to be able to adapt energy prices from renewable energy suppliers, for example, as offered with the price interface from [67].

Figure 3 shows the activity diagram of the energy agent of the instantiated MAS4AI framework, considering the game tree that has been setup by the scheduler agent.

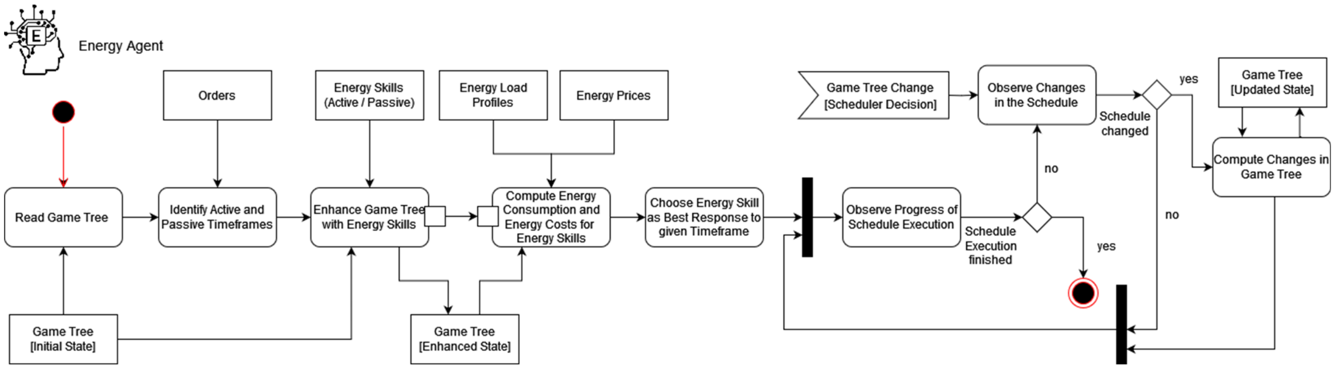

Energy Agent
Orders
Energy Skills (Active / Passive)
Energy Load Profiles
Energy Prices
Game Tree Change [Scheduler Decision]
Observe Changes in the Schedule
yes
Game Tree [Updated State]
Schedule changed
Compute Changes in Game Tree
no
Read Game Tree
Identify Active and Passive Timeframes
Enhance Game Tree with Energy Skills
Compute Energy Consumption and Energy Costs for Energy Skills
Choose Energy Skill as Best Response to given Timeframe
Observe Progress of Schedule Execution
Schedule Execution finished
no
yes
Game Tree [Initial State]
Game Tree [Enhanced State]

Figure 3. Activity diagram for energy-aware decision making of the energy agent of a CPPS.

Energy and resource agents interact with the cyber-physical parts of the system, including CPPMs and related infrastructure units. The energy agents therefore apply the energy load profiles of those production modules, which can be built with methodical data acquisition and be processed by communicating with resource agents [9]. The measured energy loads, together with the chosen skill parametrization from resource agents, can then be used to obtain the energy consumption forecast for the planned skill execution. For energy agents, the energy adaptation models of the CPPS are of interest, which formalize the degrees of freedom for each CPPM energy consumption and are formalizable as energy adaptation models. These energy adaptation models are realizable in a modular CPPS context, as are energy skills, which are also useable in a game tree for interaction with a scheduling component that can react to dynamic energy prices [16]. If the scheduler changes a part of the predefined schedule or an energy price change can be considered locally by a CPPM by choosing another parametrization, the energy agents observe this change and transfer it to the respective local decision. The related energy consumption of the energy skills is used for the reward function in the game tree to choose the best energy skill under the given and restricted timeframes as a decision from the scheduler agent.

In summary, the described elements of the system are integrated into a combined concept to realize the energy-aware scheduling adaptation in the MAS. Table 2 summarizes the main elements of the concept, divided into the main framework parts, which are the MAS, the CPPS, and the applied algorithms.

Table 2. Components of the instantiated MAS4AI framework.

MAS Framework (Janus)	Component of the CPPS (Hardware and Software)	Algorithmic Elements (Models and Solutions)
Scheduler agents	FJSSP-Scheduler (multi-objective) Game tree schedule Dynamic energy price interface	Optimization model (MILP) Game tree generator Day-ahead prices and events
Energy agents	Energy adaptation models (skills) CPPM load profile models	Game tree adaptation Energy costs computation
Resource agents	CPPMs with skill interface	-

The focus of the current contribution is on the required elements for the scheduler agents and energy agents, based on the components of the CPPS as well as the necessary algorithms. The interaction of energy agents and resource agents, considering also CPPMs with their skill interface, has already been investigated for the purpose of energy load profiling in the contribution of [9].

3.2. Design and Implementation of Scheduler Agents and Energy Agents

The concept and implementation of energy agents in the MAS4AI framework from [53] consider especially the relation with resource agents for data acquisition at the shop floor level. The research question of [9] is addressed in the current contribution, on how energy agents could interact with scheduler agents to adapt energy-optimized production schedules. Thereby, especially the design and specification between scheduler agents and energy agents are considered. Figure 4 shows the communication diagram of the interaction of a scheduler agent and an energy agent for initial scheduling and scheduling adaptation with game trees, with the instantiated framework components of Table 2.

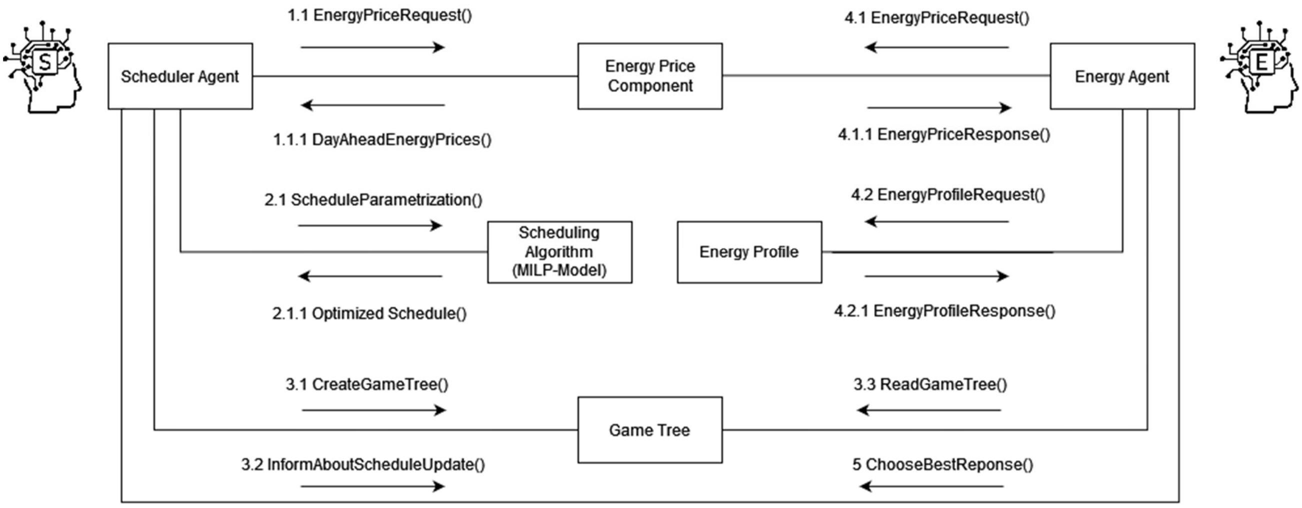

1.1 EnergyPriceRequest()
4.1 EnergyPriceRequest()
S
Scheduler Agent
Energy Price Component
Energy Agent
E
1.1.1 DayAheadEnergyPrices()
4.1.1 EnergyPriceResponse()
2.1 ScheduleParametrization()
4.2 EnergyProfileRequest()
Scheduling Algorithm (MILP-Model)
Energy Profile
2.1.1 Optimized Schedule()
4.2.1 EnergyProfileResponse()
3.1 CreateGameTree()
3.3 ReadGameTree()
Game Tree
3.2 InformAboutScheduleUpdate()
5 ChooseBestReponse()

Figure 4. Communication diagram for initial scheduling and scheduling adaptation between scheduler and energy agent with game trees.

The scheduler agent at first performs a price request to the energy price component to obtain the day-ahead energy prices. Based on those energy prices, the scheduling algorithm is performed with a defined parametrization, considering the objectives for makespan or energy costs. The optimized schedule, as a result of the initial scheduling process, is then used to create a game tree, which is overserved by energy agents. Considering the available energy profiles and energy prices, the energy agent then calculates and decides on the best response to react to the given schedule under energetical viewpoints.

The interfaces of the agents to interact with the related software components, for example, from perspective of the scheduler agent, are encapsulated and designed by using the agent skill approach in the way it is shown in [53,64]. Another possibility is the usage of the AAS to improve interoperability in the field of CPS and for integration in the context of MAS [68]. The application of the AAS for interfaces with planning or scheduling purposes is possible as well [8,69]. The interfaces for the interaction with the resource level of CPPMs follow the concept of [7]. The interface for the metering hardware to obtain energy data from a modular production system can also be designed as AAS [70] and then be used by energy agents for data acquisition [9]. The agent skills from the agent-oriented programming language SARL as presented in [65] are applied in several contributions [7,9,53,64], which are used to enter callbacks to interfaces of other systems from the programming perspective. The agent skills thus offer, beside the possibility of encapsulation, the option to reuse agent functionalities completely, or at least to a high degree, by only changing the individual programmed parts of the skill. This concept is furthermore useable to implement the required interfaces for the game tree interaction.

The formalized model of agents in combination with the behaviors and related skills with the mentioned interfaces can be found in Figure 5.

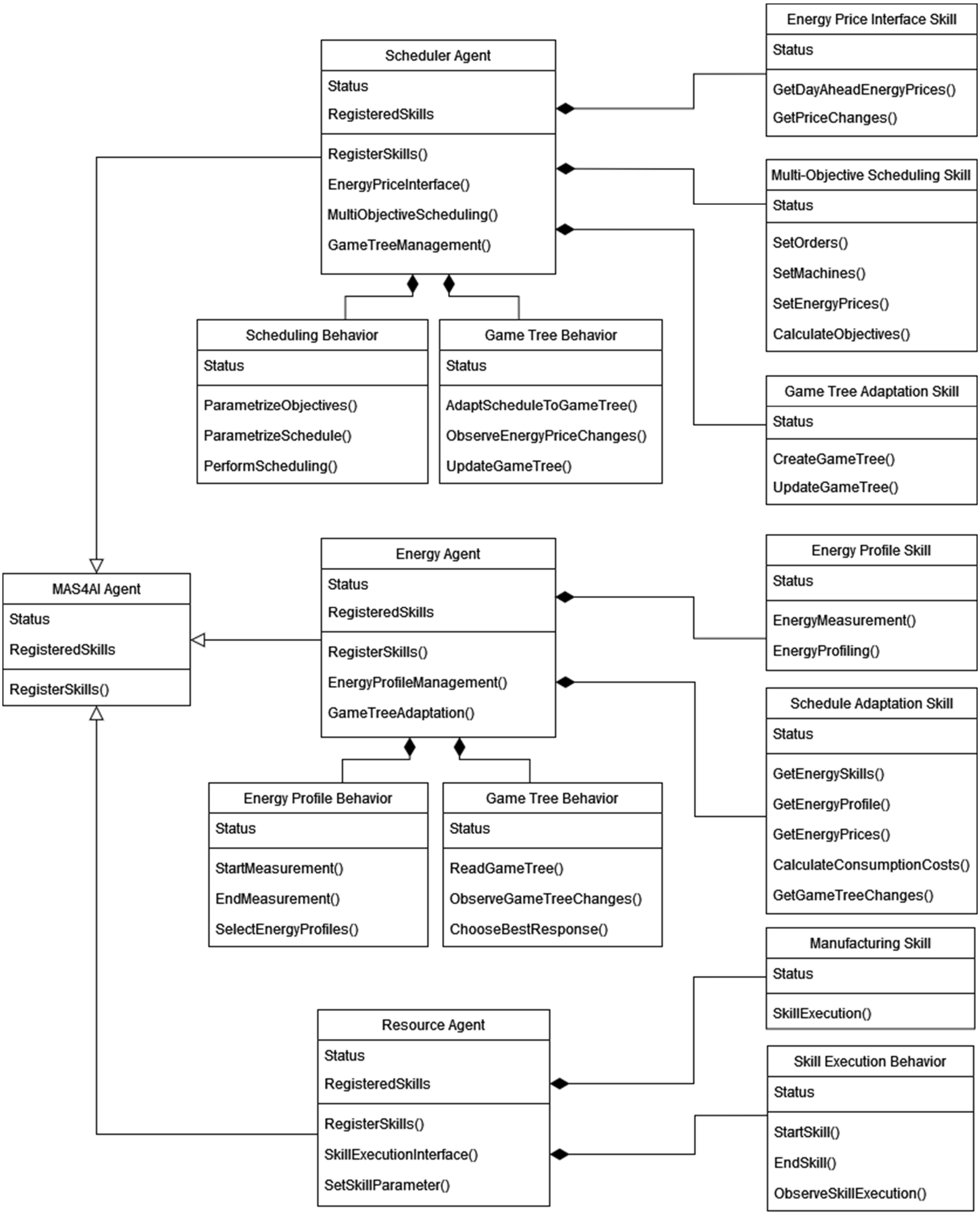


Figure 5. Class diagram with behaviors and skills of the implemented scheduler agent, energy agent and resource agent in SARL.

For the scheduler agent, at first an energy price request is performed for the energy price component. With the agent skill for initial scheduling, a MILP model for creating a schedule by using the day-ahead energy prices under multi-objective criteria can be used. The optimization can thus focus on the criteria of makespan and energy costs, based on the expected energy consumption combined with the day-ahead energy price forecast. The generation of the game tree is performed after this step, which is also encapsulated as an own skill of the scheduler agent. Another skill is to change the decisions on several parts of the game tree, if an energy price change makes it necessary to adapt the game tree from the scheduler's perspective to obtain a better result, based on the given objectives. The energy agent has the skill to respond to the game tree decisions, considering the timespan that is foreseen for production orders to be produced. Furthermore, the energy agent has access to energy profiles, which, combined with the energy adaptation models of the production modules, allow them to determine the best response to produce those orders. In this context, the functionality is important in how the scheduler agent and the energy agent are enabled to become players in the sense of game theory, using their communication to adapt the decisions of the previous generated schedule as an initial game tree with the day-ahead energy prices. If no changes are required, the agents must also acknowledge the predefined decision path. Therefore, it is important that both agent types can observe the current state of the game tree and can interpret belonging decisions.

4. Method for Energy-Aware Production Schedule Adaptation

The general model parameters and formalization are presented in Section 4.1. In Section 4.2, the general schedule adaptation mechanism as well as the necessary steps for the production schedule adaptation into game trees are described, focusing on the modeling and the transforming of an initial schedule. Section 4.3 shows the detailed rescheduling mechanism with game trees.

4.1. Model Parameters and Formalization

The formalization and the production schedule adaptation follow in general the approach of [16], so that an existing scheduling under consideration and computation of energy costs, based on day-ahead energy prices, can be transformed in an extensive-form game tree format and then used for rescheduling purposes. In general, the concept of extensive-form games can be applied to the transformed schedule. The considered modular production system can be built of several CPPMs $\mathcal{C} = \{c_1, c_2, \dots\ c_k\}$ with precedence relations. A CPPS can start the manufacturing process for a production order O_j with related machines. The processing in active and passive timeframes for an order timeframe is realized with energy skills $S_i\ \in S_{active} \cap S_{passive}$ of CPPMs, which are described in Table 3.

Table 3. Energy response skills S_i of CPPMs in $\mathcal{C}$.

Order Timeframe	Energy Response Skills S_i	Description	References
Active response S_{active}	Intensity 1 (I1)	Default skill intensity	[16,34]
	Intensity 2 (I2)	Less time and more energy than I1	[16,34]
	Intensity 3 (I3)	Less time and more energy than I2	[16,34]
	Intensity 4 (I4)	Less time and more energy than I3	[16,34]
Passive response $S_{passive}$	Idle	Default idle	[16]
	Eco-Idle (Eco)	Higher energy savings than idle	[16,71]
	Wake-on-Lan Sleep (WOL Sleep)	Higher energy savings than Eco-Idle	[16,71]

Each CPPM can produce in active timeframes in four intensities. Higher intensities of an S_{active} skill will result in a time-saving effect, but also have the effect of higher energy consumption. Passive timeframes can be adapted with a passive energy response skill $S_{passive}$ in which a deeper inactive mode also results in higher energy savings. A special element of $S_{passive}$ is the possibility of an empty response, which has no effect. This element is used to skip the selection of skills for the passive timeframes when the order processing should be performed without breaks. The applied formalization considers in general several aspects of a game for any number of players in extensive form, which are presented in [72] and based on the definitions of [73]:

- Finite set $\mathcal{N} = \{n_1, n_2, \ldots\ n_k\}$ players.
- Rooted tree, $\mathcal{T}$, as game tree.
- Leaf nodes with a tuple of payoffs, which are computed as rewards.

According to [16], each CPPM can be considered as an individual player of $\mathcal{N}$ and can use its energy response skills S_i as action space in the game tree to select and execute decisions based on the reward. The reward function for such a multi-objective schedule can be built similar to the reward functions presented in [74], and must consist of the objectives for makespan and energy costs, in which production times and idle times are formalized into the reward, for example, to avoid timeframes in which energy is expensive.

Since the schedule as well as the transformed game tree format focus on various objectives, considering makespan and energy consumption costs, the concept of an overall reward function must be adapted to several order timeframes of the schedule, to enable rescheduling of local parts. The timeframe of a production order O_j can be divided with a set $\mathcal{S}$ that consists of an active and passive timeframe tuple of the order timeframe with $(a_i, p_i) \in \mathcal{S}$. The reward R_{order} for O_j is built of the rewards for R_{active} for $(a_i) \in \mathcal{S}$ and $R_{passive}$ for $(p_i) \in \mathcal{S}$. The reward of R_{order} and the related part R_{active} should be maximized, and the value of $R_{passive}$ should be minimized, since $R_{passive}$ only contains energy costs. The reward equitation is built as follows for each production order timeframe:

$$R_{order} = R_{active} - R_{passive} \tag{1}$$

The reward for the active part consists of parameter α, which is the weight between the objectives of makespan and energy consumption costs. The value $M_{a,i,max}$ consists of the maximum makespan of $(a_i) \in \mathcal{S}$ and $E_{a,i,max}$ consists of the highest energy costs in $(a_i) \in \mathcal{S}$, built on related energy consumption and energy prices. For $E_{a,i,max}$ the highest intensity is chosen. The makespan of the decision is scaled with the maximum possible makespan for the active timeframe. For energy costs, the value is scaled with the maximum energy costs from active and passive skills. The reward R_{active} is built as follows:

$$R_{active} = (1-\alpha)\left(1 - \left(\frac{M_{a,i}}{M_{a,i,max}}\right)\right) + \alpha\left(1 - \left(\frac{E_{a,i}}{E_{a,i,max}}\right)\right) \forall (a_i, p_i) \in \mathcal{S} \tag{2}$$

The reward for the passive part considers the energy costs of $(p_i) \in \mathcal{S}$:

$$R_{passive} = \alpha\left(\frac{E_{p,i}}{E_{a,i,max}}\right) \forall (a_i, p_i) \in \mathcal{S} \tag{3}$$

4.2. General Schedule Adaptation Mechanism

The realization of an initial day-ahead schedule can be performed with various models, e.g., with a MILP model [34] or with reinforcement learning [74], in the way it is used as the pre-optimization layer proposed in [14]. These models can consist of all the required aspects to find an optimal or near-optimal solution for an initial calculated schedule, but often have the drawback of long computation times. Such a schedule can also be created by using rule-based approaches or expert knowledge. The initial schedule then specifies the possible states that should be followed in the game tree if no changes occur.

The scheduler agent provides at first the multi-objective schedule, which consists of the provided FJSSP schedule of several CPPMs that are intended to produce production orders in various intensities. An exemplary result of such a schedule with a 50:50 ratio for the objectives of makespan and energy costs with a MILP model, which can be similar designed to [34], and is based on the parameters of Section 4.1, is shown in Figure 6. The mechanism for the following scheduling adaptation into game trees follows the approach of [16], in which for each machine, passive and active timeframes are considered.

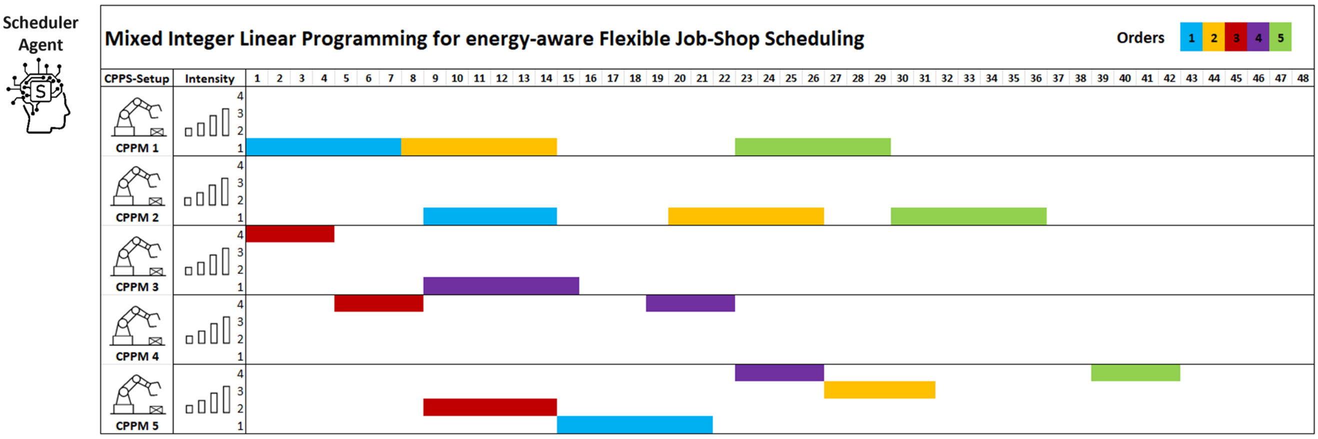


Figure 6. Exemplary calculated flexible job-shop scheduling with a MILP model.

The presented schedule adaptation logic is realized as its own component, and changes in the created game tree can autonomously be adapted with agents, fulfilling the role of the players of the game tree. Additionally, the mechanism of the timeframe consideration is extended, so that a tuple of a previous passive (p_i) and active (a_i) timeframe for each production order is always considered in combination. In Table 4, the decision-based view of each tuple is shown as an example for the visualized schedule in Figure 6.

Table 4. Decision-based view of the adapted exemplary calculated flexible job-shop scheduling.

CPPM	p_1	a_1	p_2	a_2	p_3	a_3	p_4	a_4	p_5	a_5	p_6
1	-	Intensity 1	-	Intensity 1	WOL	Intensity 1	WOL	-	-	-	-
2	WOL	Intensity 2	WOL	Intensity 1	ECO	Intensity 1	WOL	-	-	-	-
3	-	Intensity 4	ECO	Intensity 1	WOL	-	-	-	-	-	-
4	ECO	Intensity 4	WOL	Intensity 4	WOL	-	-	-	-	-	-
5	WOL	Intensity 2	-	Intensity 1	IDLE	Intensity 4	-	Intensity 3	WOL	Intensity 4	WOL

The sequence flow for the initial schedule and the energy-aware schedule adaptation are visualized in the diagram in Figure 7. The sequence starts with the daily schedule, obtaining the day-ahead energy prices from the energy price component, and performing the initial schedule in the scheduling component, which is then transformed afterwards into the game tree format. The initial schedule and its game tree are then stored and accessible for the scheduling component. In the realization stage of this schedule, when the production orders are processed, the current state of the schedule with the related progress is validated with a cyclic check for an eventual schedule adaptation.

The adaptation is intended in this case for energy price changes, which are retrieved from the energy price component. If the scheduler agent notices price changes, the related part of the game tree is identified and, if necessary, adapted by using other makespan decisions. These decisions are forwarded to the energy agent, which acknowledges those decisions by choosing the best response decision under energetical viewpoints. Following the current state of the schedule in the game tree, the scheduler agent confirms the upcoming timeframe of the schedule for the next order, which is scheduled on a production resource, and informs the related energy agent. The energy agent prepares the best response by choosing a parametrization with a suitable skill configuration, which is sent to the related resource agent for the skill execution on the respective CPPM. After the skill is performed, the schedule is updated again with the current state.

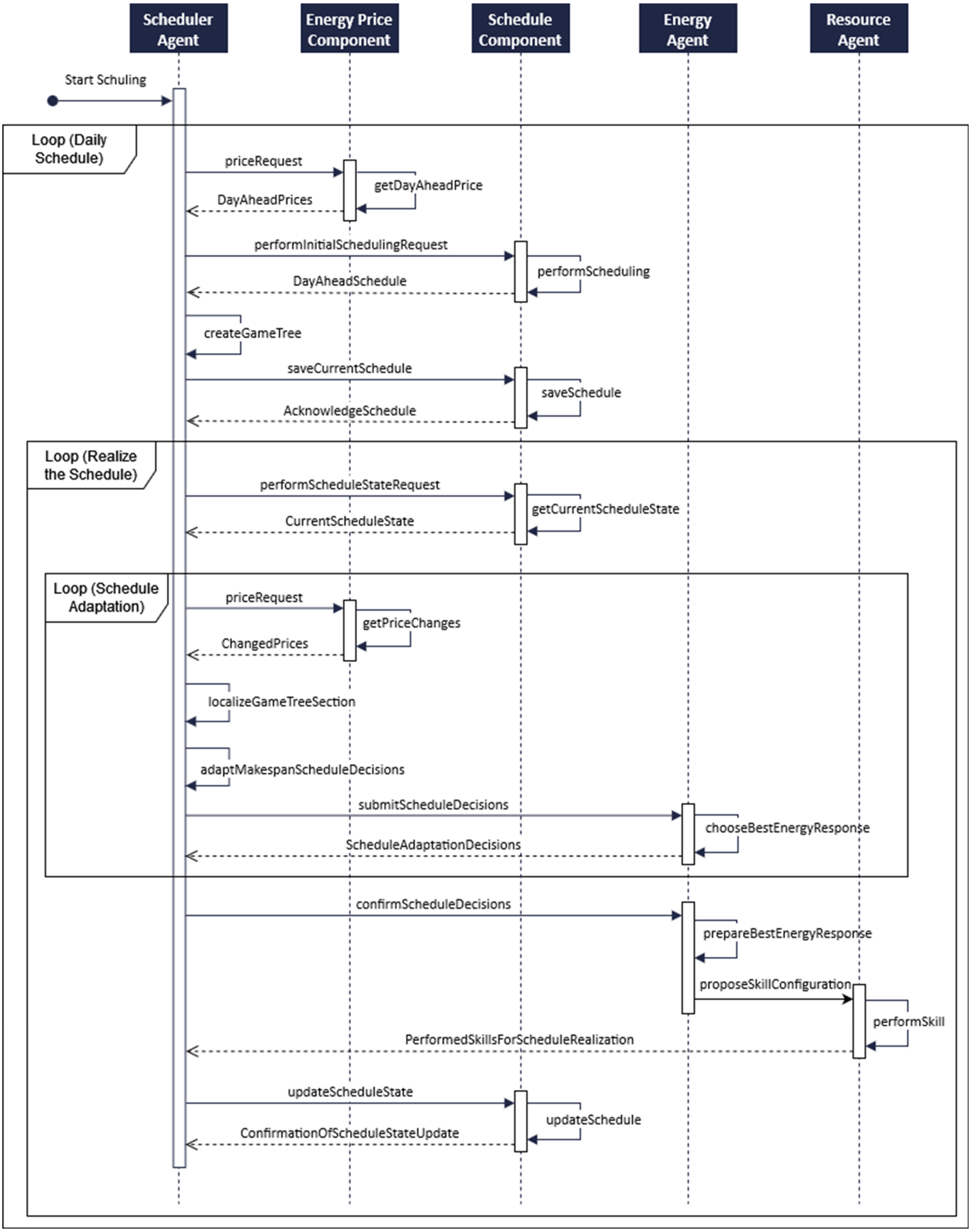
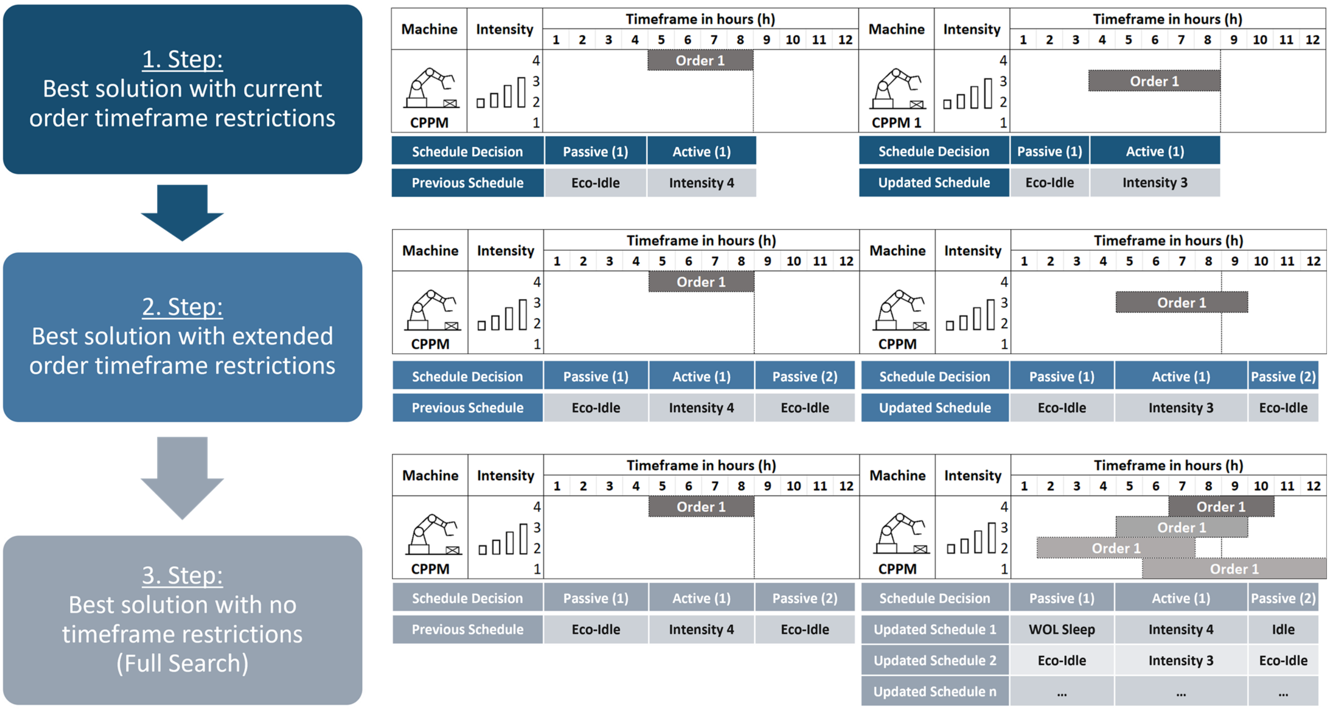


Figure 7. Sequence diagram of the agent communication to adapt energy-optimized production schedules.

4.3. Rescheduling Mechanism with Game Trees

The mechanism to adapt and change generated game trees can follow various steps, based on how the solution space of the game tree should be restricted and explored. The necessity for changes to already processed schedules is initiated by the scheduler agent. The scheduler agent observes short-term energy price changes and adapts the decisions from the previous generated day-ahead schedule by using the game tree. Energy agents of the CPPMs observe the decision changes from the scheduler agent and decide to react with their energy skills. The scheduler agent thus acts like a leader player in a Stackelberg game, in which the decision is observed by follower players and then adapted with their best response. Nevertheless, the scheduler agent in this role must also consider the expected response of the energy agents to reach the best possible response as a summarized reward for both decisions, from the scheduler agent and energy agent perspectives. The main goal of the game tree adaptation and search mechanism is to solve problems more locally and to avoid processing a full computation of the overall schedule. The mechanism for game tree search can have three steps, which are visualized in Figure 8.


1. Step:
Best solution with current order timeframe restrictions
2. Step:
Best solution with extended order timeframe restrictions
3. Step:
Best solution with no timeframe restrictions (Full Search)
Machine
Intensity
Timeframe in hours (h)
1 2 3 4 5 6 7 8 9 10 11 12
4
3
2
1
CPPM
CPPM 1
Order 1

Schedule Decision	Passive (1)	Active (1)
Previous Schedule	Eco-Idle	Intensity 4

Schedule Decision	Passive (1)	Active (1)
Updated Schedule	Eco-Idle	Intensity 3

Schedule Decision	Passive (1)	Active (1)	Passive (2)
Previous Schedule	Eco-Idle	Intensity 4	Eco-Idle

Schedule Decision	Passive (1)	Active (1)	Passive (2)
Updated Schedule	Eco-Idle	Intensity 3	Eco-Idle

Schedule Decision	Passive (1)	Active (1)	Passive (2)
Previous Schedule	Eco-Idle	Intensity 4	Eco-Idle

Schedule Decision	Passive (1)	Active (1)	Passive (2)
Updated Schedule 1	WOL Sleep	Intensity 4	Idle
Updated Schedule 2	Eco-Idle	Intensity 3	Eco-Idle
Updated Schedule n	...	...	...

Figure 8. Method for expansion of the solution space for schedule exploration with game tree search.

In general, the schedule is divided into several parts, based on the orders that should be processed by each CPPM. The reference is the given completion time for each order, so that the active processing time, combined with the previous passive timeframe, builds the solution space of an individual part of the schedule. A rescheduling is then performed for a part of the schedule if an energy price change occurs. The possible decisions for all three steps of the proposed method in general are based on the presented formalization and the degrees of freedom of the considered energy skills. These skills can be considered to adapt a rescheduling for passive and active timeframes for each order timeframe. The search mechanism in three steps is proposed to focus at first on local problem solving, following the principle of subsidiary from [24], and to extend the solution space in an iterative way.

In the first step, the completion time of each order, which is set by the scheduler agent, is considered by the energy agent as a constraint, so that the order timeframe must not be exceeded. The results of such local rescheduling can be influenced by using other energy skills, which cost less energy but result in longer processing times. Therefore, the passive timeframe before each order could be considered as well. The second step allows the consideration of an extended order timeframe by the scheduler agent, so that it is also possible to choose another order processing time, compared with the previously adapted schedule. In this step, the rescheduling decision could also reach the passive and active timeframes of the subsequent order timeframe. The constraint to search inside the combinations of the solution space is thus extended and considers additionally the first following passive order timeframe. This step also results in an adaptation of the decisions of the subsequent order and is the reason why passive timeframes are always considered in the proposed game tree model before an active order processing timeframe. For such decisions, the scheduler agents and energy agents must interact in combination for a larger part of the schedule. In the last step, better solutions are searched in the adapted game tree without constraints on the timeframe. This results in a much larger solution space since all possible combinations are searched in the game tree for the rescheduling decision.

The game tree is modeled with the Gambit software (Version 15.1) [75], which allows modeling and computation of extensive-form games by using a user interface or a Python API for the game functionalities. The game trees can be modeled in advance, imported and updated using the Gambit Python library during the runtime, or created directly programmatically. In the game tree, the rewards must be built for the makespan decision as well as for the energy decision. Energy measurements and energy load profiles are important to compute the reward of the energy costs for the timeframes and can be processed in the way described in [9,70]. Figure 9 shows a game tree example with related elements.

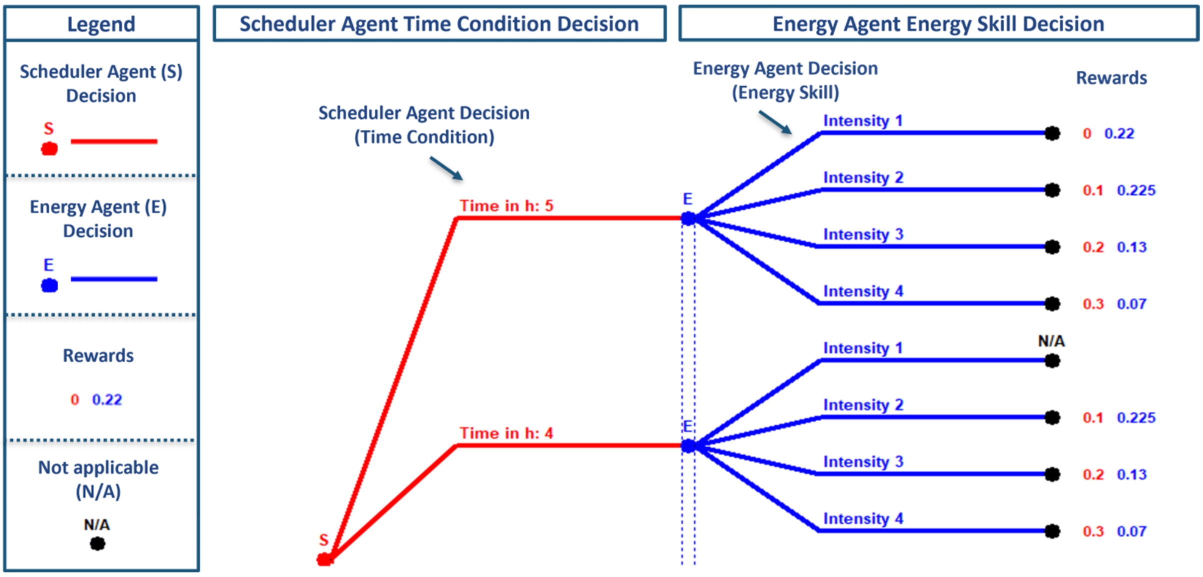


Figure 9. Game tree for an adapted part of a schedule, modeled with Gambit (Version 15.1) [75].

The scheduler agent decides in which timespan the current order should be produced, and the energy agent adapts with its best response. If a short time limit is expected by the scheduler agent to fulfill the given plan, then the selection of an intensity by the energy agent that costs more time than restricted will lead to no reward for both players. The energy agent can thus respond only with intensity, which fulfills the given timeframe constraint. In the presented example, several Nash equilibria are computed. The overall best reward for the active timeframe part of the proposed example is as well computed as Nash equilibria with the best overall reward. The scheduler agent would thus act as a Stackelberg leader player and choose the tree decision for the second fastest makespan, where the time limit in hours is four. The energy agent then chooses the energy-related skill Intensity I3 as best response. This mechanism allows it to divide and transform scheduling timeframes with multi-objective criteria into stable equilibria, where each player can consider their decisions, regarding this objective, in their best response. With the computation of equilibria based on pure strategies in extensive-form games, the best response could be computed in the Gambit software for each decision of the agents.

5. Results for Rescheduling Use Cases

5.1. Simulated Energy Price Adaptation Example

The results for the agent-based rescheduling with game trees are presented using a rescheduling use case for an energy price update for a given CPPM with simulated values. The mechanism follows the first step from Section 4.3, in which the local update of the game tree, as well as the defined solution space of a given order timeframe, is used as the solution space for the game tree search. The formalization of the game tree considers the energy-related skills of the CPPMs as machines in the model, as well as their degrees of freedom and effects on makespan and energy consumption. The objectives, makespan, and energy costs are weighted in a ratio of 50:50. For the results of the provided example, the simulation parameter settings of a CPPM are applied as shown in Table 5, similar to [16]:

Table 5. Simulation parameter settings for a CPPM.

Order Timeframe	Energy Response Skills S_i	Energy Consumption for each h	Time/Costs Effect
Active response $S_i \in S_{active}$	Intensity 1 (I1) Intensity 2 (I2) Intensity 3 (I3) Intensity 4 (I4)	40 kWh 60 kWh 100 kWh 170 kWh	5 h duration 4 h duration 3 h duration 2 h duration
Passive response $S_i \in S_{passive}$	Idle Eco-Idle (Eco) Wake-on-Lan Sleep (WOL Sleep)	5 kWh 2 kWh 1 kWh	- Costs to active: 5 kWh Costs to active: 10 kWh

To show how the mechanism of the schedule adaptation is carried out and rewards are calculated in the game tree, the day-ahead prices and event-based price updates must be considered. The energy prices for the order timeframe O_1 are shown in Table 6.

Table 6. Energy prices for the corresponding timeframe of production order O_1.

Energy Price State	Hour 1 in Order Timeframe O_1	Hour 2 in Order Timeframe O_1	Hour 3 in Order Timeframe O_1	Hour 4 in Order Timeframe O_1
Energy prices in ct/kWh Initial day-ahead forecast	6.1	6.6	9.5	10.8
Energy prices in ct/kWh Event-based price update	6.1	6.6	9.5	2.5

The rewards of the active response and passive response of the subgame tree for the given CPPM simulation parameters and day-ahead energy prices are shown in Table 7.

Table 7. Rewards of the subgame tree for the order timeframe O_1 with day-ahead energy prices.

Order Timeframe	Active Response Skills S_i	Passive Response Skills S_i	Reward R_{active} (M_a+E_a)	Reward $R_{passive}$ (E_p)	Reward R_{order} ($R_{active} - R_{passive}$)
O_1	Intensity 1 (I1)	N/A	N/A	N/A	N/A
	Intensity 2 (I2)	N/A*	0.33 (M: 0.10 + E: 0.23)	0.00	0.33
	Intensity 3 (I3)	Idle Eco-Idle WOL Sleep	0.33 (M: 0.20 + E: 0.13) 0.33 (M: 0.20 + E: 0.13) 0.33 (M: 0.20 + E: 0.13)	0.01 0.02 0.03	0.32 0.31 0.30
	Intensity 4 (I4)	Idle Eco-Idle WOL Sleep	0.32 (M: 0.30 + E: 0.02) 0.32 (M: 0.30 + E: 0.02) 0.32 (M: 0.30 + E: 0.02)	0.01 0.01 0.02	0.31 0.31 0.30

For the given example, the order must be processed within the given time limit of four hours. This decision is predefined by the scheduler agent in the game tree and must be adapted by the energy agent. The energy agent thus selects active and passive skills to react to this decision with the best response. Since Intensity I1 has a total duration for order processing of 5 h, the selection is not possible and therefore not applicable (N/A). The first selectable decision is for Intensity I2, but this intensity consumes the whole time as active processing time, so a passive timeframe is not applicable (N/A*) for this decision, as it does not exist in this combination. For this case, a special decision to skip the passive part of the related order timeframe in the game tree is selectable and applied. For intensities I3 and I4, passive skills must always be considered. The energy costs of the passive timeframes are calculated with the energy consumption of the passive response skills with the related energy price and are scaled with the highest energy costs possible in this timeframe, which is in this case the Intensity I4. This is performed to ensure the same scaling for calculating the reward for active and passive skills. Additionally, if the energy costs are weighted at zero, the passive reward becomes zero as well. Considering the rewards of all valid combinations within the subgame tree for this order timeframe and with the day-ahead energy prices, the best response would be Intensity I2 without passive energy skill. The updated rewards of the subgame tree after the energy price update are presented in Table 8.

Table 8. Rewards of the subgame tree for the order timeframe O_1 with updated energy prices.

Order Timeframe	**Active Response Skills S_i**	**Passive Response Skills S_i**	**Reward R_{active} (M_a+E_a)**	**Reward $R_{passive}$ (E_p)**	**Reward R_{order} ($R_{active} - R_{passive}$)**
O_1	Intensity 1 (I1)	N/A	N/A	N/A	N/A
	Intensity 2 (I2)	N/A*	0.25 (M: 0.10 + E: 0.15)	0.00	0.25
	Intensity 3 (I3)	Idle Eco-Idle WOL	0.26 (M: 0.20 + E: 0.06) 0.26 (M: 0.20 + E: 0.06) 0.26 (M: 0.20 + E: 0.06)	0.01 0.02 0.03	0.25 0.24 0.23
	Intensity 4 (I4)	Idle Eco-Idle WOL	0.32 (M: 0.30 + E: 0.02) 0.32 (M: 0.30 + E: 0.02) 0.32 (M: 0.30 + E: 0.02)	0.01 0.01 0.02	0.31 0.31 0.30

Since the energy price update takes place at the end of the order timeframe, the calculation of the energy costs for the passive timeframes is not affected. As a result of the price reduction, the selection of the energy-intensive Intensity I4 has lesser energy costs. The combination of Intensity I4 as response for the active timeframe in combination with Idle or Eco-Idle becomes the best response combination after the energy price update.

The adoption of the reward function to parts of the schedule is elemental in the concept of subgame trees in the overall game tree for a CPPM to compute the results for each part of the game tree. The computation of the rewards is required during the initial generation of the game tree as well as for rescheduling purposes, but in this case only for a smaller part of the game tree. The given time limit for an order timeframe is applied in the example as a constraint, which is realized as a decision about the required hours to produce an order. Skills that would lead to a longer processing time than the defined time limit have no rewards in the game tree. If a higher intensity is chosen for the active production, the time saved is considered as part of the passive timeframe before the active timeframe, so that the selection of the active response skill Intensity I3 will create in the presented example a passive timeframe in the first hour of the order timeframe, in which the passive response skills with their energy consumption can be selected and considered with energy prices.

In Figure 10, the game tree of the subgame of the presented example with the rewards from the energy price update is presented. Parts of the game tree, which are not considered due to constraints such as the selection of Intensity I1 due to the long processing time, are skipped to improve the search time in the game tree. The same logic is applied for special decisions, such as the skip of passive timeframes if they cannot occur based on the given time limit. The feasible decisions of the energy agent to react with a tuple for active and passive timeframes for the earliest realizable time limit are also available as response with the same reward for the two other scheduler decisions with lesser time limits.

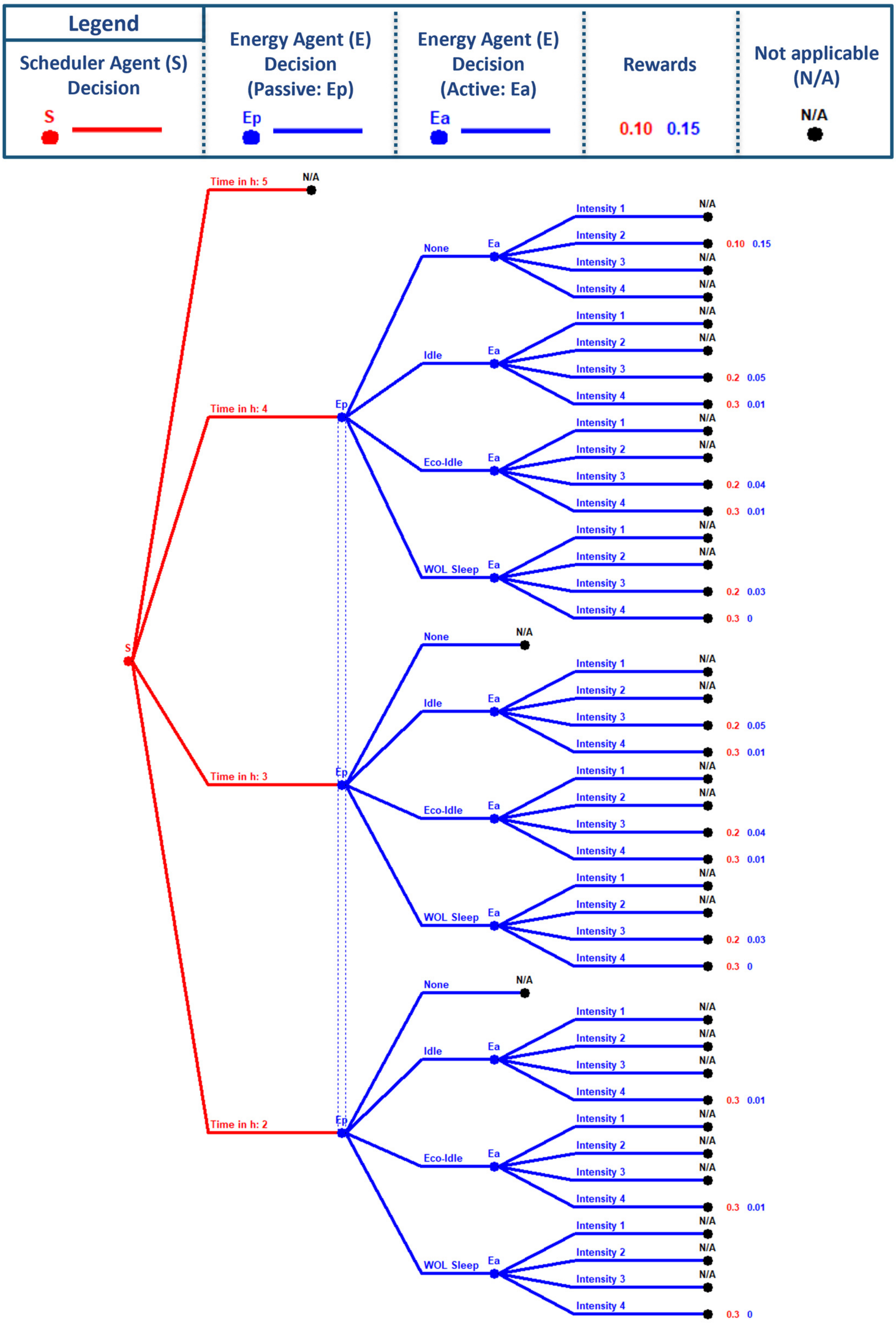


Figure 10. Game tree model of the rescheduling use case after energy price update, modeled with Gambit (Version 15.1) [75].

5.2. Prototypical Skill-Execution Adaptation

The game tree mechanism to adapt to changes can be realized and validated with different purposes for a CPPS, for example, during skill execution with CPPMs. For this purpose, the agents of the MAS can react to unforeseen events and changes on the shopfloor, which could lead to delays and disturbances to the previous schedule. The prototypical implementation was realized on a real-world demonstrator of a modular production environment for discrete manufacturing of the SmartFactory-KL. This testbed environment was also used in the prototypical implementation of [9] for energy load profiling with energy agents, realizing decision making about energy profiles during the skill execution by resource agents. The considered CPPM is shown in Figure 11 and is used in the production environment as a flexible assembly station, which provides several skills to pick up product parts from the transport system and return them if the assembly is finished. The station therefore moves the parts to a manual assembly workstation. The main component of the flexible assembly station consists of a robot arm (Universal Robots UR5e). Compared with the simulated example, the timespan and the energy consumption during a skill execution are shorter. This has an impact on the reward function of the game tree, since energy prices can be applied to this detailed level, but energy price changes are of minor importance due to the much lower timeframe and energy consumption. Nevertheless, the skill execution can be adapted using intermediate idle states as breaks during processing to react to unforeseen failures or load peaks in the system.

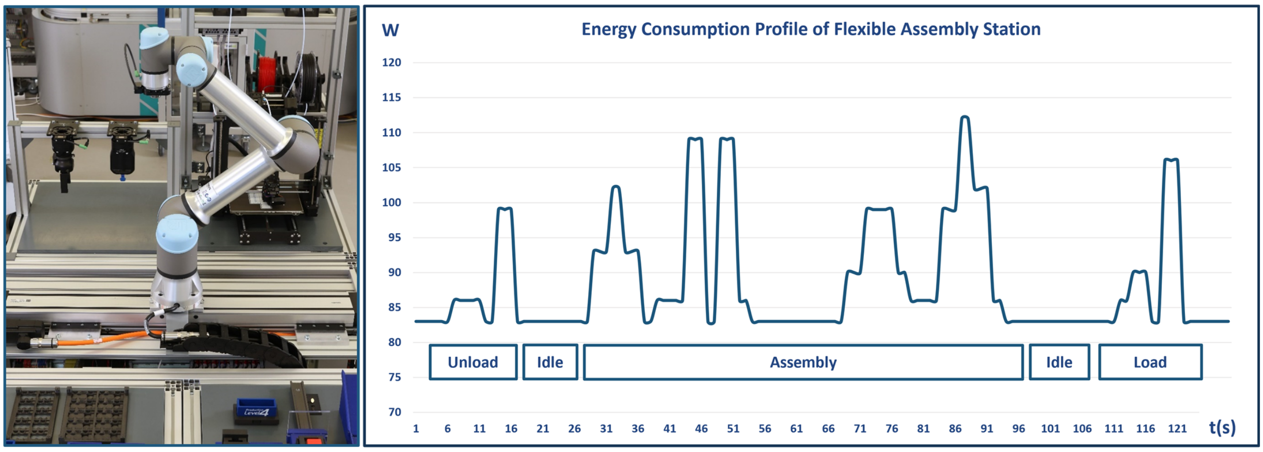


Figure 11. Flexible assembly station as CPPM of a real-world demonstrator with corresponding energy load profile.

The flexible assembly station is a CPPM with three skills, which are considered sequences in the production process. Intermediate states appear between the skill execution of two skills. Since the considered skills are not parameterizable to reach different operating speeds, the only degree of freedom to adapt to energy-related or time-related changes exists in the usage of those intermediate idle states as execution breaks. Differently from the simulated example, the focus is not on the evaluation and recalculation of the rewards of the game tree but on the selection of alternative sequences, which, if ignored, would lead to disturbances and delays in the production process. Nevertheless, the adaptability of this mechanism can be applied and validated using this example, even at such a granular level. The skill-execution sequence of the three involved skills follows a similar structure of precedence relations, which takes place not on the level between several CPPMs but inside a single CPPM with skills. The adaptation use case on this demonstrator differs between two possible events that could occur during the skill execution:

1. Prevention of energy load peaks by using idle states.
2. Consideration and handling of failures during processing.

In the skill-execution use case, the energy price is not focused on the calculation of the energy reward. Nevertheless, the energy consumption itself is considered on a detailed scale. The order timeframe to fulfill the order processing by using the CPPM is also divided into an active response and a passive response. The energy consumption of the skill execution was measured using the prototypical implementations from [9,70]. The parameter settings of the skill for picking production parts from the transport system towards the flexible assembly station of the SmartFactory-KL testbed environment are presented in Table 9.

Table 9. Parameter settings for skill execution on the flexible assembly station.

Order Timeframe	Skills S_i	Energy Consumption (Skill Execution)	Required Time (Skill Execution)	Energy Consumption (Summary)
Active response $S_i \in S_{active}$	Unload from Transport System	88.25 Ws	12 s	1059 W
Passive response $S_i \in S_{passive}$	None	0 Ws	N/A	0 W
	Idle	83 Ws	8 s	664 W
	Idle	83 Ws	12 s	996 W
	Idle	83 Ws	18 s	1494 W

Three skills of the flexible assembly station are required, which must be completed in sequence as part of the production process on this CPPM. The summarized energy consumption is built from the energy consumption and the required time during skill execution. As the passive response of the assembly station, only an idle state is useable, which always has the same energy consumption. The timeframe in which an idle state could be used is flexible, so it could be used to avoid disturbances during the skill execution. Since the flexible assembly station consists of several submodules, for example, 3D printers that can produce product parts for subsequent assembly, the possibility of energy load peaks could occur, for example, if the related 3D printer is currently heating up.

To show the possibilities of how the extensive-form games could be used in the real-world demonstrator for handling unforeseen events, the skill to pick a product part from the transport system is modeled as a game tree. The mechanism follows the same principles as the simulated use case, in which the scheduler agent provides a time condition within the production order that must be completed by the CPPM. The energy agent of the CPPM then decides on the best response and combines active response in the form of skill execution with passive response possibilities using the idle states. Additionally, the skill execution is combined with a probability, which must be observed by energy and resource agents, and which could lead to a failure, processing during an energy peak, or a successful skill execution without any issues. If a skill execution is considered in combination with a passive response in the idle state, the probability of processing during an energy peak decreases. Since no energy prices are considered, only energy consumption must be integrated into the reward function. Unlike in the simulated use case, the active response skills have no degrees of freedom in time, so the time condition from the scheduler agent does not only depends on the maximum timeframe of the active response. The reward function for a skill is always built up on the combination of makespan and energy consumption from active and passive timeframes, in which only the passive response is time flexible.

$$R_{skill} = (1-\alpha)\left(1-\left(\frac{M_{a,\,i}+M_{p,\,i}}{M_{a,i}+M_{p,\,i,max}}\right)\right)+\alpha\left(1-\left(\frac{E_{a,\,i}+E_{p,\,i}}{E_{a,i}+E_{p,\,i,max}}\right)\right)\forall(a_{\,i},p_{i})\in\mathcal{S} \tag{4}$$

The game tree rewards for the skill execution with the presented CPPM parameters and for the passive response times and the probable results, considering possible unforeseen events, are shown in Table 10. The objectives makespan and energy consumption are weighted in a ratio of 50:50.

Table 10. Rewards of the subgame tree for skill execution on the flexible assembly station.

CPPM Skill	Timeframe Condition	Passive Response Skills S_i	Probable Results	Reward R_{skill} (M_a+M_p+E_a+E_p
Unload product part from transport system	Completion in 30 s	Idle (18 s duration)	Failure Energy Peak Successful	0.00 (M: 0.00 + E: 0.00) 0.00 (M: 0.00 + E: 0.00) 0.04 (M: 0.00 + E: 0.04)
		None (direct start of the skill)	Failure Energy Peak Successful	Depends on new skill execution 0.30 (M: 0.30 + E: 0.00) 0.61 (M: 0.30 + E: 0.31)
	Completion in 20 s	Idle (8 s duration)	Failure Energy Peak Successful	0.00 (M: 0.00 + E: 0.00) 0.17 (M: 0.17 + E: 0.00) 0.48 (M: 0.17 + E: 0.31)
		None (direct start of the skill)	Failure Energy Peak Successful	0.00 (M: 0.00 + E: 0.00) 0.30 (M: 0.30 + E: 0.00) 0.61 (M: 0.30 + E: 0.31)
	Completion in 12 s	None(direct start of the skill)	Failure Energy Peak Successful	0.00 (M: 0.00 + E: 0.00) 0.30 (M: 0.30 + E: 0.00) 0.61 (M: 0.30 + E: 0.31)

The exemplary modeled game tree as an extensive-form game for probable unforeseen events is shown in Figure 12. The best rewards from the perspective of makespan and energy consumption are obtainable if the skill is directly processed without consideration of passive response skills. Nevertheless, the consideration of a passive response with an idle state reduces the probability of load peaks, so that a better adaptation to unforeseen events could be reached. If a load peak occurs, the reward for the energy agent is set to zero. If a failure during skill execution occurs, the total reward for both agents is zero. If a longer timeframe condition is chosen and a failure occurs, there is the possibility of restarting the skill execution, which leads to the possibility of a reward in the game tree.

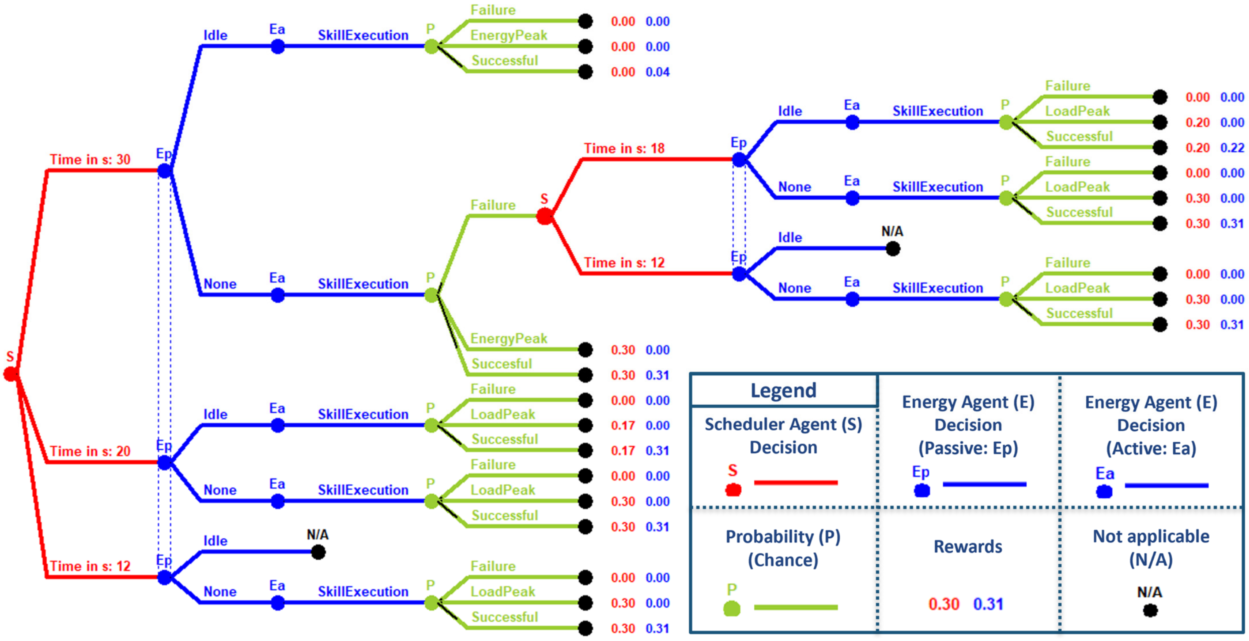


Figure 12. Game tree for the skill-execution use case, modeled with Gambit (Version 15.1) [75].

6. Discussion

The agent-based mechanism enables communication with all necessary system components to react to energy price changes and to interact with agents for an autonomous schedule adaptation. The main element of the rescheduling mechanism is the extensive-form game as a game tree. The game is setup like a Stackelberg game, in which the decisions of the scheduler agent are handled in the way of a Stackelberg leader player who moves first and whose decisions are observed and then adapted by the following agents, which are energy agents with the mechanism of Stackelberg follower players. The agent-based realization for rescheduling allows the encapsulation of functionality, such as the handling of energy profiles for individual machines or the respective parametrization of the skill interfaces by resource agents. Following the approach of the MAS4AI framework, the usage and integration of services of other system components can be realized, so that also callbacks into legacy software can be considered by using the concept of agent skills.

Through the decisions which can be modeled and observed by using the game tree, the agents can adapt to changes locally and under full information. Since local subgames of extensive-form games can have individual subgame-related Nash equilibria [56], the observability and constant update of the game tree is required for both agent types, to ensure adequate decision making for rescheduling. The presented concept is thus extendable and realizable for a various number of machines, which are considered in an initial day-ahead schedule. For each relation between a scheduler and a machine, an interaction between the scheduler agent and the related energy agent exists, so that between those agents the rescheduling takes place in an individual game tree.

The presented rescheduling use case enables the comparison of changed rewards in the game tree due to a short-term energy price update. The formalization of game trees in general enables parallel computing since each subgame tree between schedulers and energy agents could be computed and solved individually to react locally to changes. This enables a distributed computation with the best possible decision that can be found in a predefined time and is scalable with newly integrated machines. Since decision trees can contribute to human-understandable decision making [17], the game tree enables a better explainability of the rescheduling decisions. This allows the integration of humans as decision makers for rescheduling, so that humans can act in game trees in the way of agents. Since the MAS4AI framework provides the possibility for integration of various algorithms, it would be of interest for future research to choose and compare the results of various algorithmic combinations for scheduling and optimization. There is potential that biologically inspired search algorithms such as ant colony optimization or bee algorithms can contribute to this, since those algorithms also follow a multi-agent approach.

The main concept that is applied to the considered tree search is the localization of subgames so that Nash equilibria for each subgame can be computed. To avoid larger search times across the game tree, the search is performed in several steps with iterative enlargement of the solution space. A full tree search, as the last possible step on the presented solution space extension, will thus result in the drawback of long computation times, so that the main advancement of local rescheduling is lost again. Based on the main principles of the basic concept, there is potential for future extensions and applications with tree search, such as parallelization during tree search or the application of pruning rules to skip parts of the game tree that have no or most likely less influence to find a better solution. Within the solution space of single-order timeframes as well as across several subsequent-order timeframes, it can be attempted to find a better solution in a predefined computation time. The search range could then be parametrized by agents.

A main element for the realization of the simulated energy price adaptation example was the encapsulation of an overall schedule into several smaller parts, dividing the possible solution space by using the order timeframes, which consist of active and previous linked passive timeframes. The tuple of passive and active timeframes brings in flexibility to adapt longer processing times of an order into the passive times of the subsequent order timeframe during the search in an extended solution space. If a previous order is finished earlier, the time freed up can also be automatically considered in the subsequent passive timeframe. The reward functions of a multi-objective optimized day-ahead schedule, which is adapted into a game tree for scheduler agents and energy agents, can be separated for several parts of the schedule, which are adapted as individual subgames. The rewards of the leaf nodes for scheduler agents and energy agents, focusing on makespan and energy consumption costs, can be designed to result in stable Nash equilibria.

The prototypical implementation of a skill-execution adaptation was realized on a real-world demonstrator of the SmartFactory-KL, using the skills of a robot arm of a flexible assembly station. Since the timespan at this level of granularity is too short to react to energy price changes, the handling of unforeseen events and failures based on generated game trees was focused on to show the possibilities of this approach for finding and using alternative sequences. The adaptability of this approach could thus be presented during the skill execution for corresponding schedules, in which the same mechanism is applied on a more granular scale. The time condition decision from the scheduler agent can be considered by the energy agent to adapt the passive response to avoid the probability of load peaks in the CPPM or to restart the skill execution again. The realization of the prototypical implementation also showed the importance of energy measurements and energy load profiles for the calculation of the rewards in the game tree for decision making. As some energy consumptions of machines depend on different parameters besides speed and intensity, for example, temperatures in the machine itself and the environment, it is important to consider a detailed model with a range of possible energy consumptions.

In summary, both use cases show the possibilities of using extensive-form games for decision making for adaptation and rescheduling. The energy price consideration with the simulated example and energy skills shows the potential to integrate this approach with the respective agents in a MAS for autonomous energy price adaptation. Nevertheless, a drawback for practical evaluation of this use case is, that production systems with modeled degrees of freedom for electrical energy consumption rarely exist. The example was simulated to demonstrate its applicability, and the parameters of the energy skills were selected according to demonstration criteria. However, the second use case shows the possibility of applying the mechanism to a real-world demonstrator. Especially under consideration of breaks in production and different unforeseen events, further application of this mechanism can validate and further develop its robustness. For both use cases, an extension of the area of application is necessary to gain additional insights into further developments and limitations and to continue working on the open questions identified.

To place the results in the context of the contributions from the literature review, a comparison with the approaches and results of the referenced key contributions is necessary. In the game theoretical implementation of [13], an optimization for a modular production system was presented by using a potential game. The aspect mentioned for future research on the implementation of a Stackelberg game with corresponding energy agents was addressed in the scope of the current contribution. Future research can contribute to the mentioned combination in [13], using potential games for optimization across several production modules and Stackelberg games for energy-related decision making.

The realization of the prototypical adaptation presented during skill execution follows a similar aim, as shown in the contribution of [14], to have a pre-optimization layer for scheduling production units and then to react to unforeseen events by using extensive-form games, which can then be separated and calculated in several subgames with shorter solution times. The contribution in [14] presents a framework for usage and application of this mechanism in a production environment and considers aspects to improve aspects of sustainability, and where the potential for future integration into a MAS was stated.

An important aim of the current contribution was the combination of already stated research questions and that the combination of different domains and perspectives can create a new point of view for autonomous adaptation of production schedules under consideration of energy-related aspects. Based on the discussed topics, the results and limitations offer potential for future research and developments in this domain.

7. Conclusions

The concept and realization of an agent-based adaptation of energy-optimized production schedules with extensive-form games are presented in this contribution. The required components, which must be included and considered in the MAS framework, are introduced, and the activities of scheduler agents and energy agents are modeled. The design and implementation of those agent types for day-ahead scheduling, based on makespan and energy cost objectives, as well as for the agent models for realization of the scheduling adaptation with game trees, is formalized and validated with a simulated example. Additionally, the mechanism was applied to a prototypical skill-execution adaptation use case for a real-world demonstrator of the SmartFactory-KL testbed. The results of the simulated energy price adaptation use case show the realizability of agent-based rescheduling by using applied game theory with extensive-form games for short-time energy price updates. Therefore, the rewards for the energy costs of localizable subgames of the game tree, which represent production order timeframes, can be updated. The subsequent decision making focuses on local rescheduling to avoid changes due to dependencies towards a larger area of the schedule. The results of the prototypically implemented skill-execution adaptation use case show the application of the extensive-form games and how unforeseen events and failures can be modeled. Based on the rewards, which are built on measured energy consumption values, alternative decision making is calculable.

For future research, there is potential for further experiments, such as the application of various datasets for FJSSPs under energetical viewpoints or machines that have different implementations of energy skills, for example, to vary the decision options. Additionally, the consideration of constraints can be elaborated on in future research, for example, energy load limits for energy consumption during a timeframe, which is challenging if the game tree search is distributed and parallelized. Extensions of the presented game tree search can be investigated with a detailed consideration of tree pruning, for example, if cheaper energy prices are offered, to search for decisions for faster production first. Based on the successful formalization and transformation of such schedules into game trees, future research can thus focus on various approaches for faster decision making, such as investigating parallelization, tree pruning, and reinforcement learning approaches with MCTS, for example, using MCTX [76]. Additionally, there is potential for the combination of these solution concepts and benchmarking for various data constellations of such experiments. Based on the improved explainability of schedule adaptations with game trees for scheduled and executed decisions, further research can focus on the integration of human decision making using the mechanisms of game trees. The integration of humans into the MAS could, in this case, follow the general design of the MAS4AI framework [7].

Author Contributions: Conceptualization, W.M.; methodology, W.M.; software, W.M.; validation, W.M.; investigation, W.M.; writing—original draft preparation, W.M.; writing—review and editing, W.M., A.W. and M.R.; visualization, W.M.; supervision, A.W. and M.R. All authors have read and agreed to the published version of the manuscript.

Funding: This study was part of the research funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK) in the context of the project “Modulare Smart Manufacturing Gaia-X Testumgebung (SmartMA-X, 13I40V005A)”.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors would like to thank Mario Klostermeier for assistance and discussion for the formalization of the included mathematical equations, Leonhard Kunz for the helpful comments during the technical editing of the document and Michael Junker for the expertise and discussions regarding general modeling possibilities for designing activity diagrams with UML.

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

References

1. Kagermann, H.; Wahlster, W.; Helbig, J. *Recommendations for Implementing the Strategic Initiative Industrie 4.0: Final Report of the Industrie 4.0 Working Group*; Forschungsunion: Berlin, Germany, 2013.
2. Shrouf, F.; Ordieres, J.; Miragliotta, G. Smart factories in Industry 4.0: A review of the concept and of energy management approached in production based on the Internet of Things paradigm. In Proceedings of the 2014 IEEE International Conference on Industrial Engineering and Engineering Management, Selangor, Malaysia, 9–12 December 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 697–701.
3. Finn, P.; Fitzpatrick, C. Demand side management of industrial electricity consumption: Promoting the use of renewable energy through real-time pricing. *Appl. Energy* **2014**, *113*, 11–21. [CrossRef]
4. Ruskowski, M.; Herget, A.; Hermann, J.; Motsch, W.; Pahlevannejad, P.; Sidorenko, A.; Bergweiler, S.; David, A.; Plociennik, C.; Popper, J.; et al. Production Bots für Production Level 4: Skill-basierte Systeme für die Produktion der Zukunft. *Atp Mag.* **2020**, *62*, 62–71. [CrossRef]
5. Dumitrescu, R.; Westermann, T.; Falkowski, T. Autonome Systeme in der Produktion. *Ind. 4.0 Manag.* **2018**, *6*, 17–20. [CrossRef]
6. Wahlster, W. Künstliche Intelligenz als Grundlage autonomer Systeme. *Inform.-Spektrum* **2017**, *40*, 409–418. [CrossRef]
7. Sidorenko, A.; Motsch, W.; Van Bekkum, M.; Nikolakis, N.; Alexopoulos, K.; Wagner, A. The MAS4AI framework for human-centered agile and smart manufacturing. *Front. Artif. Intell.* **2023**, *6*, 1241522. [CrossRef]
8. Alexopoulos, K.; Nikolakis, N.; Bakopoulos, E.; Siatras, V.; Mavrothalassitis, P. Machine Learning Agents Augmented by Digital Twinning for Smart Production Scheduling. *IFAC-PapersOnLine* **2023**, *56*, 2963–2968. [CrossRef]
9. Motsch, W.; Simon, M.; Sidorenko, A.; Rübel, P.; Kränzler, C.; Wagner, A.; Ruskowski, M. Energy Agents for Energy Load Profiling in Modular Skill-Based Production Environments. In *International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing*; Springer Nature: Cham, Switzerland, 2023; pp. 394–408.
10. Mohsenian-Rad, A.H.; Wong, V.W.; Jatskevich, J.; Schober, R.; Leon-Garcia, A. Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid. *IEEE Trans. Smart Grid* **2010**, *1*, 320–331. [CrossRef]
11. Saghezchi, F.B.; Saghezchi, F.B.; Nascimento, A.; Rodriguez, J. Game theory and pricing strategies for demand-side management in the smart grid. In Proceedings of the 2014 9th International Symposium on Communication Systems, Networks & Digital Sign (CSNDSP), Manchester, UK, 23–25 July 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 883–887.
12. Fernández Cerero, D.; Fernández Montes González, A.; Jakóbik, A.; Kolodziej, J. Stackelberg game-based models in energy-aware cloud scheduling. In Proceedings of the ECMS 2018: 32nd European Conference on Modelling and Simulation (2018). European Council for Modelling and Simulation, Wilhelmshaven, Germany, 22–25 May 2018.
13. Schwung, D. *Maschinelle Lernalgorithmen zur Selbstoptimierung in Verteilten Produktionssystemen Basierend auf Spieltheoretischen Konzepten*; Shaker Verlag: Düren, Germany, 2021.
14. Zhang, Y.; Wang, J.; Liu, Y. Game theory based real-time multi-objective flexible job shop scheduling considering environmental impact. *J. Clean. Prod.* **2017**, *167*, 665–679. [CrossRef]
15. Bänsch, K.; Busse, J.; Meisel, F.; Rieck, J.; Scholz, S.; Volling, T.; Wichmann, M.G. Energy-aware decision support models in production environments: A systematic literature review. *Comput. Ind. Eng.* **2021**, *159*, 107456. [CrossRef]
16. Motsch, W.; Yfantis, V.; Wagner, A.; Ruskowski, M. Utilizing Extensive-Form Games for Energy-aware Production Plan Adaptation in Modular Skill-based Production Systems. *IFAC-PapersOnLine* **2023**, *56*, 2969–2975. [CrossRef]
17. Blockeel, H.; Devos, L.; Frénay, B.; Nanfack, G.; Nijssen, S. Decision trees: From efficient prediction to responsible AI. *Front. Artif. Intell.* **2023**, *6*, 1124553. [CrossRef]
18. Li, L.; Liu, H.; Wang, H.; Liu, T.; Li, W. A parallel algorithm for game tree search using gpgpu. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *26*, 2114–2127. [CrossRef]
19. Lubosch, M.; Kunath, M.; Winkler, H. Industrial scheduling with Monte Carlo tree search and machine learning. *Procedia CIRP* **2018**, *72*, 1283–1287. [CrossRef]
20. Ribeiro, L. System design and implementation principles for industry 4.0-development of cyber-physical production systems. *Stud. AB Lund* **2020**, *3*, 10–15.
21. Dafflon, B.; Moalla, N.; Ouzrout, Y. The challenges, approaches, and used techniques of CPS for manufacturing in Industry 4.0: A literature review. *Int. J. Adv. Manuf. Technol.* **2021**, *113*, 2395–2412. [CrossRef]
22. Kolberg, D.; Hermann, J.; Mohr, F.; Bertelsmeier, F.; Engler, F.; Franken, R.; Kiradjiev, P.; Pfeifer, M.; Richter, D.; Salleem, M.; et al. *SmartFactory^{KL} System Architecture for Industrie 4.0 Production Plants*; Whitepaper SF-1.2, 4; SmartFactoryKL: Kaiserslautern, Germany, 2018.
23. Ribeiro, L. Cyber-physical production systems' design challenges. In Proceedings of the 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE), Edinburgh, UK, 19–21 June 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1189–1194.
24. Bergweiler, S.; Hamm, S.; Hermann, J.; Plociennik, C.; Ruskowski, M.; Wagner, A. *Production Level 4-Der Weg zur Zukunftssicheren und Verlässlichen Produktion*; Whitepaper SF-5.1; SmartFactoryKL: Kaiserslautern, Germany, 2022.
25. Birtel, M.; Mohr, F.; Hermann, J.; Bertram, P.; Ruskowski, M. Requirements for a human-centered condition monitoring in modular production environments. *IFAC-PapersOnLine* **2018**, *51*, 909–914. [CrossRef]
26. Diedrich, C.; Belyaev, A.; Bock, J.; Grimm, S.; Hermann, J.; Klausmann, T.; Köcher, A.; Meixner, K.; Peschke, J.; Schleipen, M.; et al. *Information Model for Capabilities, Skills & Services*; Fraunhofer-Gesellschaft: München, Germany, 2022.

27. Bayha, A.; Bock, J.; Boss, B.; Diedrich, C.; Malakuti, S. *Describing Capabilities of Industrie 4.0 Components*; German Electrical and Electronics Manufacturers Association: Frankfurt am Main, Germany, 2020.
28. Kagermann, H.; Wahlster, W. Ten years of Industrie 4.0. *Sci* **2022**, *4*, 26. [CrossRef]
29. Gao, K.; Huang, Y.; Sadollah, A.; Wang, L. A review of energy-efficient scheduling in intelligent production systems. *Complex Intell. Syst.* **2020**, *6*, 237–249. [CrossRef]
30. Gahm, C.; Denz, F.; Dirr, M.; Tuma, A. Energy-efficient scheduling in manufacturing companies: A review and research framework. *Eur. J. Oper. Res.* **2016**, *248*, 744–757. [CrossRef]
31. Colangelo, E.; Hartleif, S.; Hefner, S.; Sauer, A. Energy flexibility in production planning. *Procedia CIRP* **2021**, *104*, 1095–1100. [CrossRef]
32. Motsch, W.; David, A.; Sivalingam, K.; Wagner, A.; Ruskowski, M. Approach for dynamic price-based demand side management in cyber-physical production systems. *Procedia Manuf.* **2020**, *51*, 1748–1754. [CrossRef]
33. Raileanu, S.; Anton, F.; Iatan, A.; Borangiu, T.; Anton, S.; Morariu, O. Resource scheduling based on energy consumption for sustainable manufacturing. *J. Intell. Manuf.* **2017**, *28*, 1519–1530. [CrossRef]
34. Yfantis, V.; Motsch, W.; Bach, N.; Wagner, A.; Ruskowski, M. Optimal Load Control and Scheduling through Distributed Mixed-integer Linear Programming. In Proceedings of the 2022 30th Mediterranean Conference on Control and Automation (MED), Vouliagmeni, Greece, 28 June–1 July 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 920–926.
35. Chen, G.; Zhang, L.; Arinez, J.; Biller, S. Energy-efficient production systems through schedule-based operations. *IEEE Trans. Autom. Sci. Eng.* **2012**, *10*, 27–37. [CrossRef]
36. Pach, C.; Berger, T.; Sallez, Y.; Bonte, T.; Adam, E.; Trentesaux, D. Reactive and energy-aware scheduling of flexible manufacturing systems using potential fields. *Comput. Ind.* **2014**, *65*, 434–448. [CrossRef]
37. Biel, K.; Glock, C.H. Systematic literature review of decision support models for energy-efficient production planning. *Comput. Ind. Eng.* **2016**, *101*, 243–259. [CrossRef]
38. Wooldridge, M. *An Introduction to Multiagent Systems*; John Wiley & Sons: Hoboken, NJ, USA, 2009.
39. Leitão, P. Agent-based distributed manufacturing control: A state-of-the-art survey. *Eng. Appl. Artif. Intell.* **2009**, *22*, 979–991. [CrossRef]
40. Russell, S.J.; Norvig, P. *Artificial Intelligence: A Modern Approach*; Pearson: London, UK, 2020.
41. Dorri, A.; Kanhere, S.S.; Jurdak, R. Multi-agent systems: A survey. *IEEE Access* **2018**, *6*, 28573–28593. [CrossRef]
42. Leitão, P.; Karnouskos, S. (Eds.) Industrial agents. In *Emerging Applications of Software Agents in Industry*; Elsevier: Amsterdam, The Netherlands, 2015.
43. Giret, A.; Botti, V. Holons and agents. *J. Intell. Manuf.* **2004**, *15*, 645–659. [CrossRef]
44. Van Brussel, H.; Wyns, J.; Valckenaers, P.; Bongaerts, L.; Peeters, P. Reference architecture for holonic manufacturing systems: PROSA. *Comput. Ind.* **1998**, *37*, 255–274. [CrossRef]
45. Van Leeuwen, E.H.; Norrie, D. Holons and holarchies. *Manuf. Eng.* **1997**, *76*, 86–88. [CrossRef]
46. Leitão, P.; Restivo, F. ADACOR: A holonic architecture for agile and adaptive manufacturing control. *Comput. Ind.* **2006**, *57*, 121–130. [CrossRef]
47. Derigent, W.; Cardin, O.; Trentesaux, D. Industry 4.0: Contributions of holonic manufacturing control architectures and future challenges. *J. Intell. Manuf.* **2021**, *32*, 1797–1818. [CrossRef]
48. Leitao, P.; Karnouskos, S.; Ribeiro, L.; Lee, J.; Strasser, T.; Colombo, A.W. Smart agents in industrial cyber–physical systems. *Proc. IEEE* **2016**, *104*, 1086–1101. [CrossRef]
49. Cruz Salazar, L.A.; Ryashentseva, D.; Lüder, A.; Vogel-Heuser, B. Cyber-physical production systems architecture based on multi-agent's design pattern—Comparison of selected approaches mapping four agent patterns. *Int. J. Adv. Manuf. Technol.* **2019**, *105*, 4005–4034. [CrossRef]
50. Farid, A.M.; Ribeiro, L. An axiomatic design of a multiagent reconfigurable mechatronic system architecture. *IEEE Trans. Ind. Inform.* **2015**, *11*, 1142–1155. [CrossRef]
51. Giret, A.; Trentesaux, D.; Salido, M.A.; Garcia, E.; Adam, E. A holonic multi-agent methodology to design sustainable intelligent manufacturing control systems. *J. Clean. Prod.* **2017**, *167*, 1370–1386. [CrossRef]
52. Vogel-Heuser, B.; Salazar Cruz, L.A.; Ryashentseva, D.; Ocker, F.; Hoffmann, M.; Brehm, R.; Bruce-Boye, C.; Redder, M.; Lüder, A. Agentenmuster für flexible und rekonfigurierbare Industrie 4.0/CPS-Automatisierungs-bzw. Energiesysteme. In *VDI-Kongress Automation 2018*; VDI: Baden, Germany, 2018.
53. Motsch, W.; Sidorenko, A.; Jungbluth, S.; Hengel, K.; Wagner, A. *Smart Factory Testbed Setup–Final Results*; Zenodo: Geneva, Switzerland, 2023.
54. Nash, J. Non-cooperative games. *Ann. Math.* **1951**, *54*, 286–295. [CrossRef]
55. Holler, M.J.; Illing, G.; Napel, S. *Einführung in die Spieltheorie*; Springer: Berlin/Heidelberg, Germany, 1991; Volume 3.
56. Selten, R. Spieltheoretische Behandlung eines Oligopolmodells mit Nachfrageträgheit: Teil i: Bestimmung des dynamischen Preisgleichgewichts. *Z. Für Die Gesamte Staatswiss./J. Institutional Theor. Econ.* **1965**, *121*, 301–324.
57. Fudenberg, D.; Tirole, J. *Game Theory*; MIT Press: Cambridge, MA, USA, 1991.
58. Saqlain, M.; Ali, S.; Lee, J.Y. A Monte-Carlo tree search algorithm for the flexible job-shop scheduling in manufacturing systems. *Flex. Serv. Manuf. J.* **2023**, *35*, 548–571. [CrossRef]

59. Browne, C.B.; Powley, E.; Whitehouse, D.; Lucas, S.M.; Cowling, P.I.; Rohlfshagen, P.; Tavener, S.; Perez, D.; Samothrakis, S.; Colton, S. A survey of monte carlo tree search methods. *IEEE Trans. Comput. Intell. AI Games* **2012**, *4*, 1–43. [CrossRef]
60. Schaeffer, J. Distributed game-tree searching. *J. Parallel Distrib. Comput.* **1989**, *6*, 90–114. [CrossRef]
61. Feldmann, R.; Monien, B.; Mysliwietz, P.; Vornberger, O. Distributed game tree search. In *Parallel Algorithms for Machine Intelligence and Vision*; Springer: New York, NY, USA, 1990; pp. 66–101.
62. Holcomb, S.D.; Porter, W.K.; Ault, S.V.; Mao, G.; Wang, J. Overview on deepmind and its alphago zero ai. In Proceedings of the 2018 International Conference on Big Data and Education, Honolulu, HI, USA, 9–11 March 2018; pp. 67–71.
63. Fu, M.C. AlphaGo and Monte Carlo tree search: The simulation optimization perspective. In Proceedings of the 2016 Winter Simulation Conference (WSC), Washington, DC, USA, 11–14 December 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 659–670.
64. Sidorenko, A.; Motsch, W.; Wagner, A. *User Manuals on Accessing and Using the MAS*; Zenodo: Geneva, Switzerland, 2022.
65. Rodriguez, S.; Gaud, N.; Galland, S. SARL: A general-purpose agent-oriented programming language. In Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), Warsaw, Poland, 11–14 August 2014; IEEE: Piscataway, NJ, USA, 2014; Volume 3, pp. 103–110.
66. Teiwes, H.; Blume, S.; Herrmann, C.; Rössinger, M.; Thiede, S. Energy load profile analysis on machine level. *Procedia CIRP* **2018**, *69*, 271–276. [CrossRef]
67. aWATTar Deutschland GmbH. Tarif Hourly. 2024. Available online: https://www.awattar.de/tariffs/hourly (accessed on 7 April 2024).
68. Rongen, S.; Nikolova, N.; van der Pas, M. Modelling with AAS and RDF in Industry 4.0. *Comput. Ind.* **2023**, *148*, 103910. [CrossRef]
69. Siatras, V.; Mavrothalassitis, P.; Bakopoulos, E.; Nikolakis, N.; Alexopoulos, K. *Modelling of the Planning Agent (Version 1)*; Zenodo: Geneva, Switzerland, 2022. [CrossRef]
70. Motsch, W.; Sidorenko, A.; David, A.; Rübel, P.; Wagner, A.; Ruskowski, M. Electrical energy consumption interface in modular skill-based production systems with the asset administration shell. *Procedia Manuf.* **2021**, *55*, 535–542. [CrossRef]
71. Profibus Nutzerorganisation e.V. (2020). OPC UA for Energy Management. Available online: https://de.profibus.com/downloads/opc-ua-for-energy-management-companion-specification (accessed on 7 April 2024).
72. Hart, S. Games in extensive and strategic forms. *Handb. Game Theory Econ. Appl.* **1992**, *1*, 19–40.
73. Kuhn, H.W. Extensive games and the problem of information. In *Contributions to the Theory of Games*; Princeton University Press: Princeton, NJ, USA, 1953; p. 193.
74. Felder, M.; Trat, M.; Ovtcharova, J. *Energy-Flexible Job-Shop Scheduling Using Deep Reinforcement Learning*; Gottfried Wilhelm Leibniz Universität Hannover: Hannover, Germany, 2023.
75. Savani, R.; Turocy, T.L. Gambit: The Package for Computation in Game Theory, Version 15.1. 2024. Available online: http://www.gambit-project.org (accessed on 10 April 2024).
76. Babuschkin, I.; Baumli, K.; Bell, A.; Bhupatiraju, S.; Bruce, J.; Buchlovsky, P.; Budden, D.; Cai, T.; Clark, A.; Danihelka, I.; et al. The DeepMind JAX Ecosystem, 2020. 2010. Available online: http://github.com/deepmind (accessed on 10 April 2024).

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