

Article

Towards Energy Efficient Scheduling of Manufacturing Systems through Collaboration between Cyber Physical Production and Energy Systems

Maroua Nouiri ^{1,2,*}, Damien Trentesaux ¹  and Abdelghani Bekrar ¹ 

¹ LAMIH, UMR CNRS 8201, Université Polytechnique Hauts-de-France, 59313 Valenciennes, France; damien.trentesaux@uphf.fr (D.T.); abdelghani.bekrar@uphf.fr (A.B.)

² LS2N UMR CNRS 6004, Université de Nantes, 2 Avenue du Prof. Jean Rouxel, 44475 Carquefou, France

* Correspondence: maroua.nouiri@uphf.fr

Received: 31 July 2019; Accepted: 19 November 2019; Published: 22 November 2019



Abstract: Currently, enhancing sustainability, and in particular reducing energy consumption, is a huge challenge for manufacturing enterprises. The vision of the fourth industrial revolution (so-called “industry 4.0”) is not only to optimize production and minimize costs, but also to reduce energy consumption and enhance product life-cycle management. To address this challenge, a multi-agent architecture aimed at elaborating predictive and reactive energy-efficient scheduling through collaboration between cyber physical production and energy systems is proposed in this paper. Smart, sustainable decision tools for cyber physical production systems (CPPS) and cyber physical energy systems (CPES) are proposed. The decision tools are data-driven, agent-based models with dynamic interaction. The main aim of agent behaviours in the cyber part of CPPS is to find a predictive and reactive energy-efficient schedule. The role of agents in CPES is to control the energy consumption of connected factories and switch between the different renewable energy sources. Dynamic mechanisms in CPPS and CPES are proposed to adjust the energy consumption of production systems based on the availability of the renewable energy. The proposed approach was validated on a physically distributed architecture using networked embedded systems and real-time data sharing from connected sensors in each cyber physical systems. A series of instances inspired from the literature were tested to assess the performance of the proposed method. The results prove the efficiency of the proposed approach in adapting the energy consumption of connected factories based on a real-time energy threshold.

Keywords: predictive scheduling; reactive scheduling; energy efficiency; multi-agent modelling; cyber physical production system; cyber physical energy system; renewable energies; particle swarm optimization

1. Introduction

Over the last few years, the fourth industrial revolution has attracted more and more attention worldwide. Industry 4.0, or smart manufacturing, combines the strengths of traditional industries with new cutting edge technologies [1]. It is the combination of several technological developments that embrace both products and processes. Manufacturing systems are converted into intelligent connected factories using new technologies such as cloud manufacturing, internet of things (IoT), cyber physical systems (CPS), big data analytics (BDA), and information and communication technologies (ICT). Networking, digitalization, and computing are keys to achieving sustainable goals for such systems. Future production, energy, transport, and logistics systems should exhibit adaptive performances

such as flexibility, efficiency, reliability, security, and sustainability [2–5]. The new concepts of CPS and internet of things (IoT) both play a key role in addressing these social and governance challenges. CPS are considered as innovative technologies capable of handling both the physical processes and computational aspects of interconnected systems [6]. Using the internet as a means of communication, CPS integrate software and computational aspects with storage aspects based on mechanical and electronic functionalities [1]. The IoT is presented as a dynamic and distributed environment including various smart devices that are able to interact with other devices in their environment [7]. The idea of the IoT is to enable the inter-connectivity of things, ideally with any other device in any environment, and at any time using any available communication service [8]. Typical resources (machine, operator) are converted into intelligent objects and smart humans so that they become able to sense, act, and behave within a smart connected environment [9]. However, it is necessary to develop a new control methodology that can manage the data and information communicated while defining appropriate protocols for negotiation/collaboration and reaction, when confronted with changes in the environment (machine breakdown, new job arrivals, variation in processing time, fluctuation in energy, etc.). Improving productivity and resource efficiency should result in a decrease in the consumption of raw materials and energy.

With the advent of renewable energies, the global energy system is undergoing a profound transformation. To find the most suitable areas to install wind farms, Konstantinos et al. [10] proposed a methodology framework that considers multiple criteria that can affect the location. In the same context, Ioannou et al. [11] developed a spatial decision support system to find the most suitable locations for power plant installations by combining two techniques: multi-criteria decision analysis and fuzzy systems. The performance of both methodologies was validated on real-life case studies. Nevertheless, the availability of this type of energy is highly variable and difficult to predict [12]. Thus, ensuring the best use of resources and developing new more sustainable ways of producing and using energy is crucial.

Our work, which relies on this context of sustainable development, exploits the concept of CPS and uses IoT technologies to deploy an energy-efficient distributed control architecture based on coordination between cyber physical production systems (CPPS) and cyber physical energy systems (CPES). As decentralized control is one of the main principles in industry 4.0 systems, in this work a multi-agent architecture, named EasySched, is proposed and implemented on a physically distributed architecture composed of embedded systems. *“The increasing volatility and unpredictability of energy availability, supply, and cost require the integration of highly reactive behaviour in control laws”* [13]. Thus, collaborative, predictive, and reactive mechanisms are proposed to provide energy efficient scheduling decisions. The mechanisms concern the calculation (predictive) and the adjustment (reactive) of the order of production of goods according to the availability of renewable energy.

Renewable energy production is difficult to predict because it is highly dependent on uncontrollable conditions (e.g., weather conditions). The data shared between CPS producing goods and CPS producing renewable energy is used to adjust scheduling decisions while ensuring sustainable goals. To illustrate this reactivity, developing a multi-agent system in a host (e.g., a personal computer) through a framework is insufficient. For this reason, the EasySched architecture was validated through multiple experiments on physically distributed systems. The innovative validation of this architecture was conducted in a comprehensive, physically distributed way, using networked embedded systems.

As first proof of concept, the factories and the energy providers were represented by embedded systems and sensors, respectively. The sensors were used to simulate energy perturbations and all the components were connected via a Local Area Network (LAN) network. To assess our proposal, the energy sources considered were wind and photovoltaic, the production systems were job-shops, and the energy consumption was adjusted by changing the speed of the production machines. Our approach dynamically adapts production in real time, and, therefore, energy consumption to energy supply. Note that this work is a proof of concept. When applied on a realistic case, some

parameters should be adapted. A mathematical model is possible to confirm the results obtained by our approach and the Constraint Programming (CP) optimizer.

A literature review on cyber physical systems and energy efficiency scheduling methods is presented in Section 2. Details of the proposed architecture are given in Section 3. The implementation of EasySched is detailed in Sections 4 and 5 provides the results and the interpretations of the experiments. A conclusion and future prospects are described in Section 6.

2. Literature Review

The scientific issue in this paper is about sustainable production and more precisely the design of “energy-efficient manufacturing scheduling systems” in the context of industry 4.0 [12]. Research on energy-efficient scheduling methods has been actively developed. Concurrently, with the rapid development of CPS technologies, research on the implementation of CPS to design energy-efficient methods has been continued [14]. The literature review is divided into two parts. The first one discusses research on cyber physical production and energy systems to achieve sustainability. The second part presents and analyses the state-of-the-art in energy efficient static and dynamic scheduling methods.

2.1. CPPS and CPES

Several researchers have reviewed articles about trends, issues, activities, and applications of industry 4.0 [5,9]. For example, Liao et al. [15] analyzed academic articles on the topic of Industry 4.0, summarized research activities, and indicated potential research directions. The detailed reviews of industry 4.0 indicate that CPS and IoT technology are receiving increasing attention from the community. In this context, many different kinds of CPS can be defined. Figure 1 illustrates some industrial domains in which the concept of CPS is used. In this paper, we consider the two following types of CPS:

- **Cyber Physical Production System (CPPS):** defined “as a system of systems comprising autonomous and cooperative elements connecting with each other in situation dependent ways on and across all levels of production, from processes through machines up to production and logistics networks, enhancing the decision-making process in real time, in response to unforeseen conditions” [16].
- **Cyber Physical Energy System (CPES):** to promote building energy management, “researchers investigated the possibility of integrating energy management systems with Cyber-Physical Systems to form Energy-Cyber-Physical Systems” [17]. CPES are considered as systems of systems with dynamical behaviour through interactions between different complex components and systems. CPES combine domains from physical energy systems such as electric power grids, and distributed energy resources with information and communication infrastructures (ICT) [18].

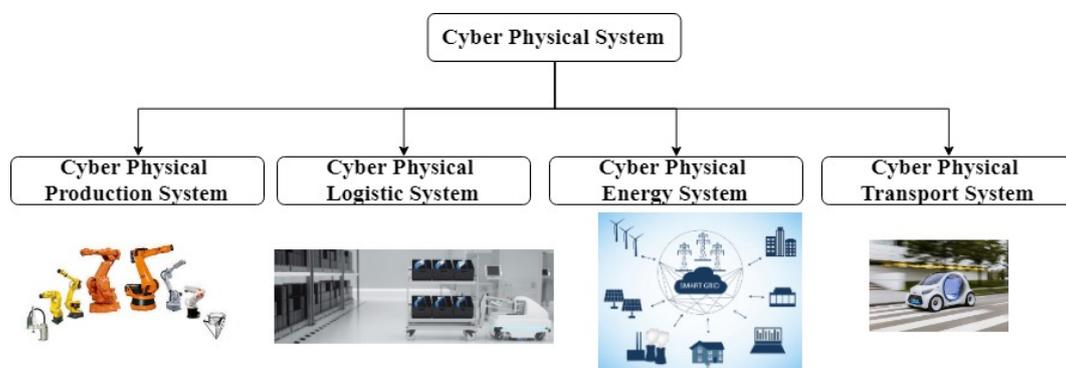


Figure 1. Examples of cyber physical systems.

Different methodologies and approaches have been used to address CPPS that are classified and detailed in the work of Cardin et al. [19] through an analysis framework. For example, in the context of intelligent manufacturing, the authors in [20] propose a CPS architecture for the shop

floor considering various aspects such as interconnectivity between devices, multi-sourcing in data acquisition and smart decision-making and learning. Keil [21] designed an autonomous control system for decentralized CPPS manufacturing systems. The performance of the design was validated on a case study of a semiconductor production system. In a survey of CPES [22], Macana et al. identified the main research areas and challenges in energy systems. The authors in [23] proposed a dynamic battery model comprising different states representing discharge/recovery effects with scheduling algorithms. Yang [24] studied the application of IoT in a smart grid and highlights the main issues relating to large-scale IoT infrastructures in co-simulation platforms. In [25], the authors proposed a real time analysis method to manage energy in CPES for micro-grids. The proposed analysis verifies the feasibility of the scheduling. The performance and effectiveness of the method was validated through experimental results. For instance in [26], a micro-grid is considered as a CPES that is able to manage the distributed power and energy storage devices dynamically. Wu et al. [27] develop a real-time energy management system to increase the use of solar energy to avoid over-charging and over-discharging of batteries in the house; this prolongs the life of the batteries in order to meet the electricity needs of a smart building.

CPS technology can be combined with scheduling algorithms to design energy-efficient scheduling solutions. For example, in an energy-efficient job shop context, Feng et al. [14] developed a dynamic scheduling system based on a cyber physical energy monitoring system. The proposed system uses a modified genetic algorithm with multi-layer coding to ensure the energy-efficiency of the solution. A CPS was used to monitor the energy consumption of a manufacturing process. In the next section, scheduling methods dealing with energy are discussed.

2.2. Energy Efficient Scheduling Methods

Energy efficient (or green) methods are used to tackle various sustainability challenges. They must be integrated into the CPPS control systems and must interact with the CPES supplying their energy. Contributions relevant to energy efficient scheduling methods found in the literature can be classified into two main categories described hereinafter.

2.2.1. Static Energy Efficient Scheduling Methods

The first category groups energy efficient static scheduling methods without considering perturbations. For instance, the authors in [28] present a framework to solve a multi-objective optimization problem that minimizes total energy consumption and total tardiness. Fang et al. proposed a mathematical programming model of the flow shop scheduling problem that considers peak power load, energy consumption, and the associated carbon footprint in addition to cycle time [29]. Authors in [30] suggested a new approach to optimize energy consumption in a flexible job shop taking into account resources selection and the order of operations. They also proposed a mixed integer programming model to minimize both the makespan and the energy used. In addition to these two objectives, other studies also consider an additional objective, which is the tardiness. For instance, in the context of energy-aware scheduling, the authors in [31] developed a centralized-distributed model and a mathematical model that considers these three objectives.

A mixed integer linear programming (MILP) mathematical model is formulated in [32] for the energy-oriented hybrid flow shop scheduling to minimize both the tardiness and the non-processing energy. The authors also developed an evolutionary-based algorithm to solve the problem. A three-stage decomposition method was suggested by authors in [33] to minimize the energy used in the flow-shop scheduling problem. In the flexible job shop context, a non-dominated sorting genetic algorithm (NSGA)-based scheduling approach was suggested by Zhang et al. [34] to minimize the energy used and the makespan.

Many other studies focused on minimizing CO₂ emissions. For example, Li and Cao [35] proposed a two-stage model to minimize CO₂ emissions using two different strategies. In a similar context, the authors in [36], proposed an efficient multi-objective approach based on a genetic algorithm in

order to minimize total carbon emissions and total late work. By minimizing the energy used, the carbon emission and production costs are also minimized especially at peak times when the use of energy increases or when electricity costs become very high. In this context, Moon et al. in [37] proposed a method to minimize total production costs for the flexible job-shop scheduling problem while considering perturbations in energy sources and storage. Mikhaylidi et al. in [38] suggested an approach based on dynamic programming to generate an energy consumption schedule that indicates both the time and the amount of electrical energy used by each operation at each time period. In the context of sustainability in manufacturing systems, an evaluation and improvement approach was suggested by Liu et al. [39] to enhance the performance of the production system and the resources efficiency while considering environmental constraints.

The common point in the research cited above is that the schedule must consider another important feature, which is the energy consumed when processing jobs in addition to classical key performance indicators such as makespan, tardiness, and workload. The energy consumption (EC) can be integrated as an objective or a constraint, and can refer to machine energy consumption, CO₂ emissions, energy prices, etc.

2.2.2. Dynamic Energy Efficient Scheduling Methods

Taking unexpected disruptions into account increases the complexity of the scheduling problem. The authors in [40] classified the different approaches used to deal with disruptions into proactive, reactive, and hybrid methods. The predictive-reactive hybrid method is one of the most studied approaches. A predictive schedule is first generated to be executed until a disruption occurs. Then, a repair technique is launched to handle failures. Rescheduling is one of the repair procedures used in the reactive part of a hybrid approach. Rescheduling is a procedure that aims to adapt an optimal production schedule to dynamic and stochastic environments [41].

The rescheduling problem is of great importance. Researchers have focused on profitability and sustainability to construct a new class of rescheduling methods namely energy-efficient rescheduling (EER) methods [42]. Other studies have focused on the rescheduling problem in a dynamic job-shop environment in which the production resources work at various rates. For instance, Salido et al. [43] proposed a new match-up approach with a memetic algorithm to minimize the total energy used.

Many researchers have addressed the dynamic scheduling problem in flexible manufacturing systems (FMS). For example, in the context of energy consumption, the authors in [44] developed a new mathematical model based on goal programming to minimize the energy used while considering the efficiency of the schedule generated. The authors also proposed a genetic algorithm-based approach for rescheduling. Nouri et al. [42] proposed a green rescheduling method (GRM), an extended version of the work in Nouri et al. [45]. The GRM takes into account energy efficiency when solving the flexible job shop rescheduling problem. The above mentioned studies propose energy-efficient scheduling methods considering traditional impromptu disruptions such as due date changes, machine breakdowns, and random job arrivals. The dynamic variation of energy availability is not considered.

A few studies have focused on energy fluctuation as a source of internal perturbations in manufacturing scheduling systems. For instance, Pach et al. [46] proposed a potential field-based control model to switch resources on/off reactively depending on the situation of the FMS to reduce energy wastage. Overall, power consumption control was introduced to maintain this consumption below a dynamically determined energy threshold. Trentesaux et al. [47] presented a set of key requirements when designing multi agent or holonic architectures (MAS/HMS) for future energy aware production scheduling systems. The authors discuss the importance of design approaches that must be able to observe and control variables of energy consuming resources, and support the reconfiguration of the manufacturing system according to the cost and availability of energy as well as the availability of manufacturing resources. A predictive and reactive multi-agent architecture integrating energy consumers and energy producers is specified, however no model is proposed. Inspired from [47], Nouri et al. [48] proposed a multi-agent-energy-aware production scheduling

and rescheduling system based on decision sharing and distribution. Negotiation protocols were proposed and validated to find predictive sustainable production solutions. However, no reactive decision mechanism was provided.

Table 1 summarizes the related studies in energy rescheduling, specifying the type of perturbation, the method, and the objective.

Table 1. Research most related to the problem studied.

Authors	Research Problem	Type of Perturbation	Objective	Solution Approaches
He et al. [30]	Flexible job-shop	Without perturbation	Minimize energy consumption and makespan	Nested partitions algorithm, mixed integer linear programming (MILP) model
Tonelli et al. [31]	Energy aware scheduling	Without perturbation	Minimize job tardiness and energy consumption	Centralized-distributed model, mathematical model
Li et al. [35]	Job-shop manufacturing system	Without perturbation	Minimize CO ₂ emissions	Two stage carbon efficiency model
Piroozfard et al. [36]	Multi-objective flexible job shop scheduling problem	Without perturbation	Minimize total carbon footprint and total late work	Improved multi-objective genetic algorithm
Moon et al. [37]	Flexible job-shop scheduling problem with perturbations	Without perturbation	Minimize total production costs	Constraint programming, mixed-integer programming
Nouiri et al. [42]	Flexible job-shop rescheduling problem	Machine breakdown	Minimize energy consumption and makespan	Predictive reactive green rescheduling method (GRM) based on particle swarm optimization (PSO)
Salido et al. [43]	Dynamic job-shop rescheduling	Machine disruption	Minimize energy consumption	New match-up technique, memetic algorithm
Zhang et al. [44]	Dynamic scheduling problem	Resource and job related disruptions	Minimize energy consumption, improve schedule efficiency	Goal programming mathematical model, genetic algorithm
Pach et al. [46]	Flexible manufacturing systems	Energy	Minimize energy wastage	Potential field-based control model
Nouiri et al. [48]	Energy aware scheduling and rescheduling systems	Machine disruptions, variation of processing time and energy	Find predictive sustainable solutions	Predictive reactive multi-agent architecture

As environmental impacts become a growing concern worldwide, the power-provisioning problem is a challenge for many designers. In the domain of data centre design, Li et al. 2012 [49] proposed a power management scheme called “iSwitch” that aims to balance the charge of renewable energy power load between different power supplies. They consider one type of renewable energy (wind) and the supplies represent the server clusters (i.e., VMs). To generate an optimal bidding strategy for a virtual power plant, Riveros et al. in [50] proposed an approach that considers perturbations in terms of electrical energy and energy costs in the market using stochastic optimization by evaluating various bidding strategies. Fan et al. [14] addressed the effect of disruptions on the reliability of power systems. They developed a rescheduling approach to find the most suitable scheduling solution by reducing the variances of branch power flows. The proposed method was evaluated and validated on a real-case study.

To sum up, energy-efficient scheduling research with objectives relating to environmental influences (energy consumption, CO₂ emissions, waste, etc.) is increasing. However, energy-efficient rescheduling methods taking the fluctuation of renewable energy as a source of perturbation into account are limited in the context of manufacturing. State of the art scheduling methods are based mainly on optimization tools (meta-heuristic) and they integrate simulation tools helping to maximize

production efficiency and offset environmental effects. A mathematical model (MILP) considering the environmental performance as a constraint or as an objective is then formulated. Few studies have used a dynamic energy threshold to achieve sustainable production, whereas the majority of studies have applied prior-experimental data (a fixed energy threshold).

A large number of investigations on energy-aware manufacturing methods have been conducted to improve the energy-efficiency of production in the context of the fourth industrial revolution. However, the validation of the research already mentioned ignores the real-time events on the shop floor. Contributions also remain conceptual with little attention paid to practical validations. Finally, from our review, we noticed that few studies elaborate on the coordination and collaboration between CPPS and CPES to achieve production sustainability.

To address these shortcomings, this paper proposes a multi agent architecture, named EasySched, to schedule the production of goods in a predictive and reactive way. The architecture is based on collaboration between CPPS and CPES. The sustainable decision tools are based on a distributed control model based on an intelligent solving method (meta-heuristic/heuristic for scheduling and rescheduling) using a multi agent system coupled with IoT technology that enables devices (CPPS and CPES) to connect and collaborate with each other and act as an entire system.

As real-time scheduling plays an important role in CPS [25], it is primordial to propose new reactive mechanisms to handle the energy consumption. Two rescheduling methods are also proposed to take into account fluctuations in renewable energy in real time. EasySched was validated in a completely distributed way, integrating hardware and software. Reactive mechanisms are proposed to adapt the consumption of production factories according to an energy threshold communicated in real time by CPES from energy providers. The proposed architecture is detailed in the next section.

3. Contribution

This section provides a description of our contribution. Firstly, the design approach of “EasySched” is presented. Then, the reactive mechanisms and the implementation of the hardware are detailed. The validation tests concern the predictive and reactive parts.

3.1. Design Approach

Solving the scheduling problem consists in assigning and organizing the execution of a set of tasks on a set of machines and specifying the start and end time of each operation. The solution or “schedule” is characterized by a specific performance indicator (makespan, tardiness, etc.). However in the case of sustainable production, environmental performance indicators such as energy consumption, CO₂ emissions, and percentage waste reduction must be considered when constructing the schedule. In this work the makespan and the energy consumption of resources processing jobs are considered as the temporal and environmental performances respectively. The schedule must also respect several constraints. These constraints can be classified according to their deterministic (precedence constraint for example) or dynamic nature, taking into account the availability of machines over the planning horizon, for example. The predictive schedule represents the off-line solution calculated based on prior-experimental data (operating time, energy consumption, etc.), whereas the reactive one is formulated on-line using the real data and real situation of the production environment (considering unpredicted events). The constraints cited above should be considered as the design stage of the scheduling system. Several conceptual design methods have been proposed to ensure the most global view of the implications of the sustainability principles at different levels of the manufacturing system. For instance, references [51–53] proposed tools and conceptual models that help the developer to design and implement sustainable intelligent systems (SIMS). To ease the implementation of SIMS, multi-agent systems are suitable for distributed decision-making models enabled by a distributed agent-based architecture. In this work, the proposed multi-agent architecture was designed according to the Go-Green design methodology ANEMONA (A Multi-agent Methodology for Holonic Manufacturing Systems) [53]. The design system is completely distributed to facilitate responsiveness to a variety

of unforeseen events. The Go-Green ANEMONA is based principally on the concept of Go-Green manufacturing holon proposed in [47]. In addition to the classical effectiveness-oriented mechanisms, the Go-Green holon considers efficiency-oriented mechanisms (optimizing sustainability means) to make decisions and/or execute operations. The details of the proposed architecture, including the description of its agents, are provided in the next section.

3.2. Description of the Proposed Architecture

The proposed hybrid, EasySched, combines predictive and reactive decisions and is based on cooperation between CPPS and CPES. Figure 2 represents a general view of the architecture, specifying the different types of energy suppliers and energy consumers that can be considered. The energy-consuming production systems can be a single machine, a flow shop, or a flexible manufacturing system, etc. Wind power and solar power are considered as renewable energy sources. The IoT manufacturing gateway enables the manufacturing objects to sense, interconnect, and interact with each other to automatically and adaptively carry out manufacturing logics (see Figure 2). Different possible connections can be designed: human-to-human, human-to-machine, and machine-to-machine cooperation [5]. The cyber components refer to the sensing, communication and decisional computing systems, whereas the physical components comprise the factories with renewable energy generators. The architecture is based on the coordination processes between factories producing goods and energy providers to find a sustainable decision. The intelligent decision reflects either a schedule or reschedule in SIMS and a switching/distribution order in smart energy systems. Sustainable decisions are obtained by smart sustainable decision tools in CPPS and CPES. Both tools are data-driven agent model based on optimization algorithms with dynamic internal interactions with manufacturing objects (CPPS, cyber human system (CHS), enterprise resource planning (ERP), etc.) and external interactions with other manufacturing factories or energy providers (a set of CPES). The smart sustainable tools of CPES and CPPS are controlled by the energy scheduler agent (ESA) and factory scheduler agent (FSA) respectively. Figure 3 illustrates the agent model of EasySched based on communication between FSA and ESA. The FSA is mainly responsible for:

- Running meta-heuristics to find an energy-efficient predictive schedule.
- Controlling local events.
- Rescheduling and negotiation capabilities.

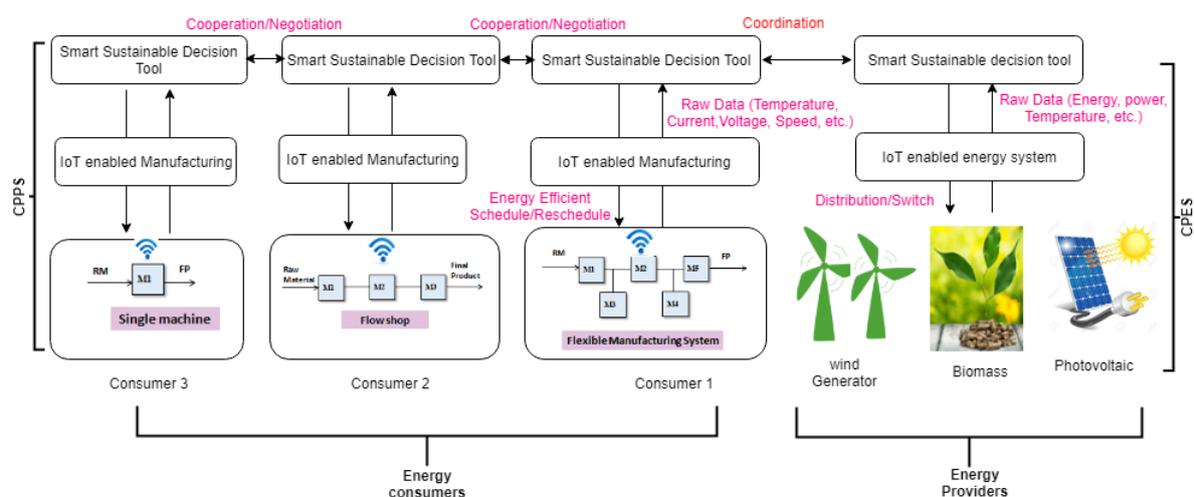


Figure 2. A general view of the EasySched Architecture.

The ESA is mainly responsible for:

- Validating predictive solutions for factories
- Controlling energy consumption of connected factories.

- Switching between renewable energy sources.
- Negotiation capabilities.

Both FSA and ESA have predictive and reactive roles. They interact with each other and/or depend on each other. The details of the smart sustainable decision tools of CPPS and CPES are described hereinafter.

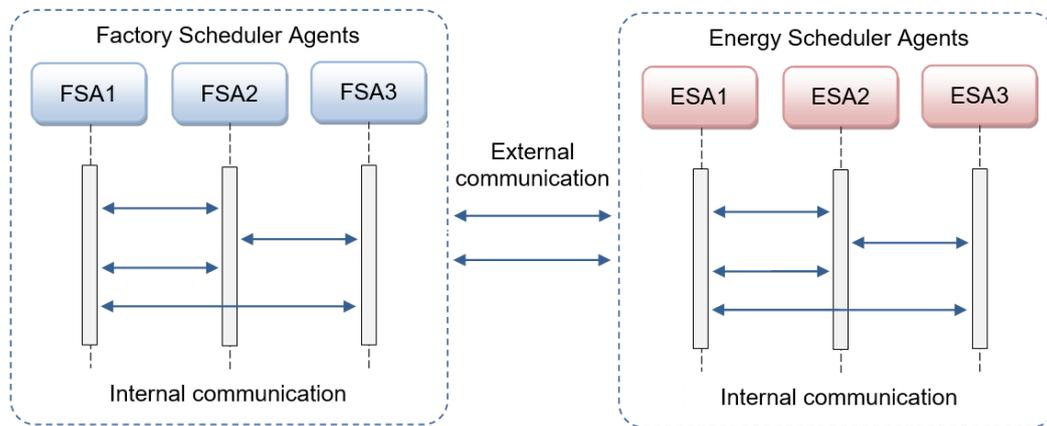


Figure 3. Agent-based control architecture of EasySched.

3.3. Smart Sustainable CPES Decision Tool

The cyber part of CPES is comprised of a smart sustainable decision tool integrating the embedded computing level and embedded communication skills (IoT technology) (see Figure 4). The smart tool is a data-driven agent based model with dynamic interactions and it is controlled by the energy scheduler agent (ESA). As mentioned before, the EasySched is hybrid, and thus the roles of the ESA are predictive and reactive.

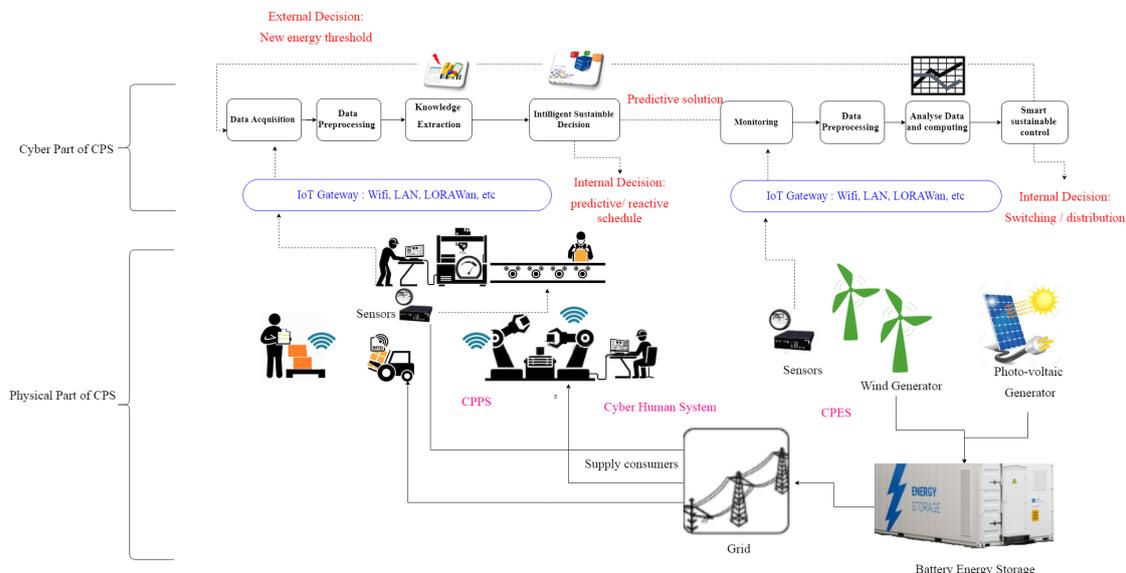


Figure 4. A detailed view on EasySched: Coordination between cyber physical production systems (CPPS) and cyber physical energy systems (CPES). Cyber physical systems (CPS).

- **Predictive role:** This role validates the predictive schedules of factories. In fact, the ESA agent receives external data from CPPS representing an energy demand that reflects the energy consumption of the predictive solution calculated by the cyber part of the CPPS. The CPES approves the solution if it satisfies the energy demands, otherwise, it refuses. More details of the

predictive behaviour are provided in [48].

- **Reactive role:** This on-line role controls and monitors, in real time, the availability of renewable energy and especially dynamic events. To achieve this goal, the reactive behaviours of the energy scheduler agent ESA are monitoring data, pre-processing data, analyzing data, and computing as follows:
 - **Monitoring data:** Data acquisition via sensors is an important aspect of industry 4.0 or smart systems in general. Sensors are the machine’s gateway to sensing its surrounding physical environment. Using appropriate sensor installation in CPES, various signals such as vibration, pressure, or temperature can be extracted. This behavior is cyclic and repeated every fixed period P1. Raw data such as temperature, electric power, voltage, and energy consumption are collected and saved to be processed. The collected data is useful to find a good sustainable decision for maintenance, control, scheduling, etc.
 - **Pre-processing data:** Data pre-processing behaviour is conducted on the acquired data. This includes checkout, debugging, duplicate checking, and duplicate removal, producing useful data [20].
 - **Analysing data and computing:** The aim of this behaviour is to analyze the process and transform the raw data collected into smart data that can be used to make decisions. First, the ESA checks if there is a false alarm or not before making the decision. This behaviour is cyclic and repeated every period P2 (where $P2 > P1$). As mentioned in Algorithm 1, if there is a fluctuation in energy, a new energy threshold is calculated by the ESA and a rescheduling decision order is sent to the factory scheduler agent (FSA). The rescheduling order specifies the rescheduling time and the energy constraint to be respected. The reception of this type of message implies that the FSA must execute a rescheduling technique to obtain a new schedule adapted to the changes imposed by the energy suppliers. If no disturbance is detected, the ESA sends a message indicating the absence of energy fluctuation to the FSA agents.

Algorithm 1 Pseudo code of energy scheduler agent (ESA) alarm triggering behaviour (Period P2)

```

1: if Alarm = "Perturbation detected" then
2:   calculate new energy Threshold
3:   Reschedule.Energy = new energy Threshold
4:   Reschedule.time = P2
5:   ControlMessage.Object = Reschedule
6:   Send (ControlMessage, FSA)
7:    $P2 = x \times P2$ 
8: else
9:   ControlMessage.State = "No energy perturbation"
10:  Send (ControlMessage, FSA)
11: end if

```

Our proposed architecture takes into consideration two situations relevant to the configuration of energy providers.

- Situation 1: Each factory is connected to a specific renewable energy generator (wind, photovoltaic, etc.) controlled by a specific ESA. In other words, each ESA controls a specific source of energy (wind for example) and the factories using this source.
- Situation 2: All renewable energy generators are combined and controlled by the same ESA. In this situation, two weighting parameters α , β are used to control the power supply portion of each source depending on its availability and/or the cost price of the energy type. The weighting

parameters are numbers between [0, 1] that indicate the degree of importance of the energy type on the energy supply (see Equation (1)).

$$Power = \alpha P_{Wind} + \beta P_{photovoltaic}, \quad \text{where } \alpha + \beta = 1. \quad (1)$$

The data collected is used to determine α and β dynamically. If, for example, there is maintenance of a wind generator, α is set to 0 and all the factories will be supplied by the photovoltaic energy supply.

According to the energy available and the efficiency of each renewable source, it is necessary to switch the consumption between the different energy sources. Thus, an intelligent algorithm must be used to enable the optimal combination between sources according to the relation in Equation (1). α and β define the rate of power used from wind and photo-voltaic sources respectively. The configuration of the weighting parameters should take into account: (1) the threshold energy reserve that must be in the batteries, (2) the duration of consumption, (3) the price, and (4) the power demand at any time.

Consequently, the CPES decides locally to switch from one source to another. Algorithm 2 illustrates the proposed switching control behaviour of ESA when two kinds of renewable energy are available (wind, photovoltaic, etc.).

Algorithm 2 Pseudo code of ESA switching control behaviour (Period P2)

```

1: read  $P_{Wind}$  and  $P_{photovoltaic}$ 
2: if  $P_{Wind} > P_{photovoltaic}$  then
3:    $\alpha > \beta$ 
4: else
5:   if  $P_{Wind} < P_{photovoltaic}$  then
6:      $\alpha < \beta$ 
7:   else
8:     if  $P_{Wind} == 0$  then
9:        $\alpha = 0$  and  $\beta = 1$ 
10:    else
11:       $\alpha = 1$  and  $\beta = 0$ 
12:    end if
13:  end if
14: end if

```

3.4. Smart Sustainable CPPS Decision Tool

As we can see from Figure 4, CPPS implementation for the shop floor consists in integrating components such as sensors, Radio Frequency Identification (RFID) devices, and measurement devices in the manufacturing resources and distributing them in the production environment. The smart sustainable decision tool of CPPS is a data-driven agent-based model with optimization algorithms and dynamic interactions. The factory scheduler agent possesses the following predictive and reactive roles:

- **Predictive role:** This off-line role consists of calculating a predictive schedule that minimizes makespan and energy consumption. The predictive solution is determined by a particle swarm optimization (PSO) algorithm proposed in [54] where a particle represents a potential solution to the problem. Each of these particles is characterized by a position vector, a velocity that allows the particle to move and a neighbourhood, a set of particles (neighbours), that interact directly with the particle, in particular that which has the best criterion. Equation (2) represents the bi-objective function used in PSO to evaluate the solutions. It combines both the makespan (C_{max}) and the total energy consumption (E_{tot}). The two objectives are not of the same magnitude or units,

therefore, a normalization phase is required. A weighting factor γ makes it possible to determine the degree of importance of each objective.

$$F = \min \gamma \frac{C_{max}}{Maxmakespan} + (1 - \gamma) \frac{E_{tot}}{MaxEnergy}. \quad (2)$$

The predictive decision of CPPS aims to find an energy efficient schedule. The FSA sends an energy demand to the ESA. If the later refuses the demand, the PSO algorithm is executed again but with a new value γ found after a negotiation phase with other CPPSs of other factories. The weighting parameter is reduced by a value λ favoring the reduction of the energy consumption. The pseudo code of negotiation protocol is detailed in Figure 3. Algorithm 3 presents the pseudo code of negotiation protocol. For more details of the negotiation protocol between CPPS are given in [48].

Algorithm 3 Pseudo code of negotiation protocol between factory scheduler agents (FSAs)

```

1: receive rescheduling need trigger
2: send proposal ( $\lambda$  value)
3: if FSA accept the proposal then
4:   execute PSO with the current  $\lambda$  value
5: else
6:    $\lambda = \lambda - 1$ 
7:   execute PSO with new value of  $\lambda$ 
8: end if

```

- **Reactive role:** The on-line objective of the FSA is to monitor and detect internal disruptions and then react by executing an intelligent rescheduling method with interactions with CPPS and CPES. To achieve this goal, the CPPS also has monitoring, pre-processing, and computing behaviors like the CPES. However the reactive decision concerns calculating a new reschedule while respecting a new energy threshold constraint. These behaviours are detailed as follows:
 - **Monitoring data:** The FSA acquires the raw data (e.g., temperature, electric power, speed, new job arrivals, and machine breakdowns) from internal CPPS or external data that represents a new energy threshold received in real time from the CPES.
 - **Pre-processing data:** Similar to CPES behaviour; it aims to structure the raw data collected.
 - **Computing:** This reactive behaviour aims to make a sustainable decision based on the data collected (sensor data, internal event (internal disruptions), and external events (fluctuation of energy from CPES)). A rescheduling method is launched to find a new schedule. The pseudo code of the reactive behaviour of the FSA agent is given in Algorithm 4. This periodic behaviour is repeated every period $P3$ (where $P3 > P2$). If the message received from ESA does not refer to any perturbation, the scheduling is applied according to the original plan. Otherwise, a rescheduling technique is executed. In fact, a rescheduling optimization method is triggered to generate an updated schedule if working conditions vary significantly (internal, external) and re-scheduling adjustments are necessary. The proposed rescheduling methods are described in the next subsection.

Algorithm 4 Pseudo code of external perturbation control behaviour of FSA (Period P3)

```

1: Reception ControlMessage from Energy scheduler agent ESA
2: if ControlMessage == "No energy perturbation" then
3:   Continue scheduling based on predictive solution
4: else
5:   Extract affected subparticle ( Reschedule.timeResch)
6:   Apply a rescheduling method on the subparticle to find a new schedule that respects the energy
   threshold Reschedule.tauxEnergy
7: end if

```

3.5. Rescheduling Methods in the Smart Sustainable Tool of CPPS

Dynamic events in a production system significantly influence the quality and efficiency of the predictive solution. Several rescheduling methods have been proposed to address this problem. Three main groups were found: right-shift rescheduling, total rescheduling, and partial rescheduling. Examples include affected operation rescheduling (AOR) proposed by Vieira et al. [55] and modified AOR (mAOR) proposed by Subramaniam and Raheja [56]. However they do not take into account the sustainable aspect. This will also impose radical changes in the way scheduling repairing approaches are designed and implemented [42]. Before presenting the proposed rescheduling techniques, the type of perturbation considered is detailed. Figure 5 illustrates the information needed to react to a perturbation. This information concerns the affected resource, as well as the date of occurrence, the duration, and the type of disruption (e.g., repairable or un-repairable). The affected resource can be either reusable or consumable. Uncertainties in terms of machine disruptions, variations in processing time, new job arrivals, and fluctuations in renewable energy are considered in our approach. A specific technique is executed depending on the type of disruption. In this work, two rescheduling techniques are proposed to address fluctuations in renewable energy considered as sources of disruption in the system.

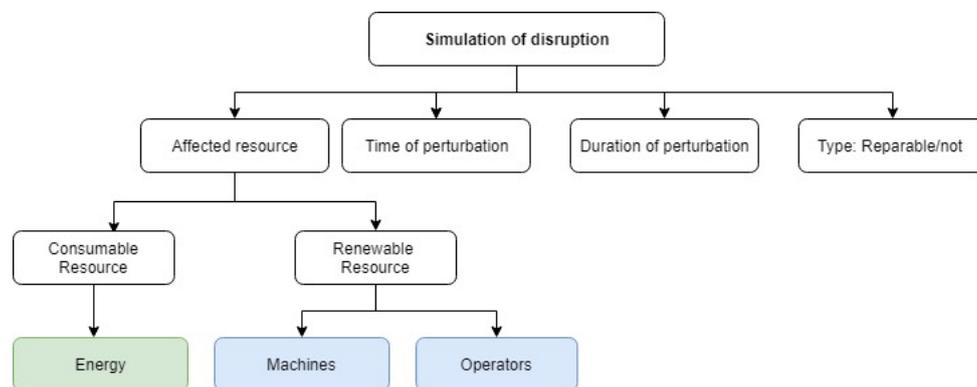


Figure 5. Simulation of disruption.

3.5.1. Rescheduling Method 1

The objective of the rescheduling method, executed by the FSA, is to find a new schedule that respects the new energy threshold communicated in real time by the CPES. In this work, the factories of the manufacturing system are job shops with machines working at different speeds. The principle of the rescheduling method is firstly to determine the operations affected in the predictive solution. The speed of the machines of affected operations is then reduced to reach a new energy consumption that respects the energy threshold. The machines are key manufacturing resources as well as the main energy sources in a workshop, therefore, energy scheduling is usually conducted to effectively manage machines from an energy consumption perspective [34]. In this work, the adjustable variable was the speed of the machines. Figure 6 summarizes the flow chart of the method. In each iteration,

the speed of an affected operation is reduced and the energy consumption (EC) is calculated. If the new EC respects the threshold, the process is finished. Otherwise, the process is repeated on all affected operations. If in the final iteration the constraint is not respected, a penalty is associated to the reactive schedule. In this case the second rescheduling technique is used.

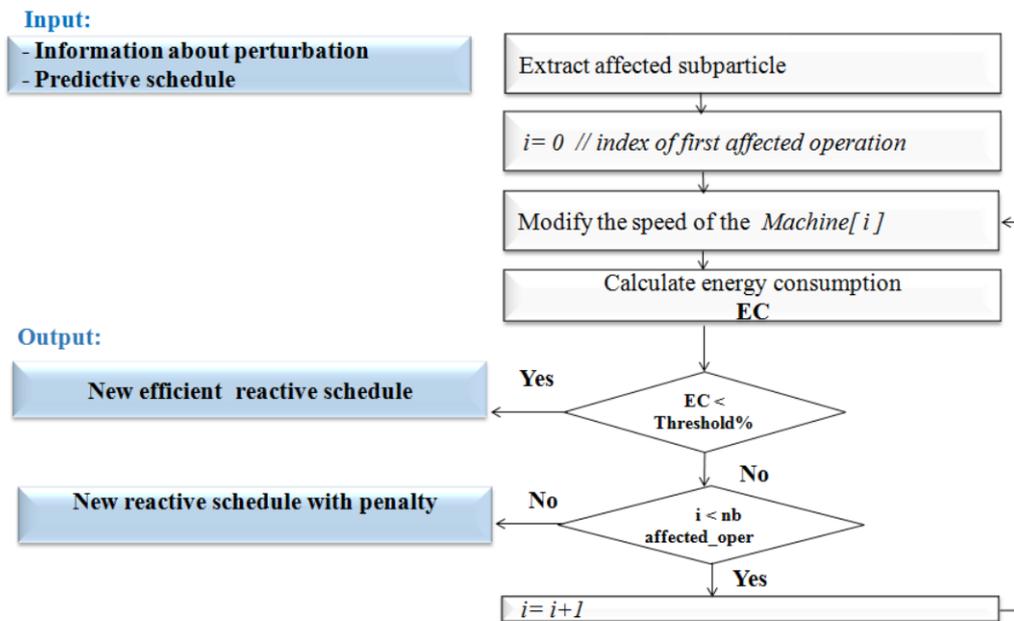


Figure 6. Flow chart of the first rescheduling method.

3.5.2. Rescheduling Method 2

The second rescheduling method is based on changing of sequence of the affected operations, which was unchanged in the previous method. However, the stability of the schedule is likely to be different compared to the predictive solution. The main idea of the method is to permute the operations while reducing the speed of the machines. In each iteration, the new energy consumption (EC) is calculated and the best solution is selected. Figure 7 presents the flow chart of the method.

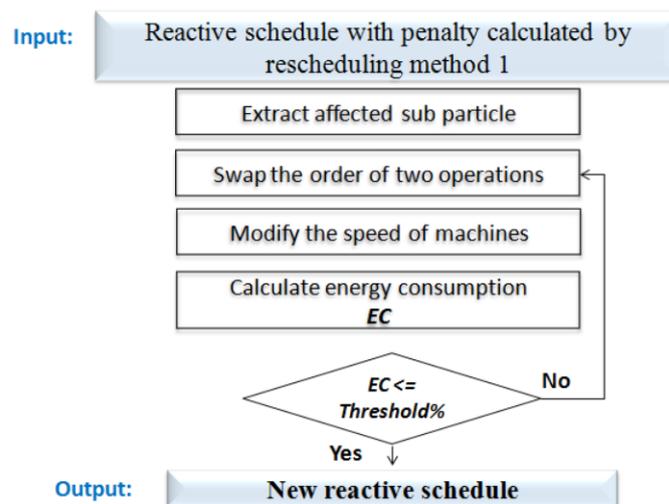


Figure 7. Flow chart of the second rescheduling method.

4. Implementation of EasySched

This section details the physical distributed systems used as an example to verify the feasibility of the proposed architecture and to assess the performance of EasySched. A physical distributed validation, on different systems, integrating the physical elements (hardware with connected sensors (IoT)) and the software is proposed in this article. These elements are connected via networks and cooperate to provide sustainable decisions. The real target system is the AIP-PRIMECA cell at the Polytechnic University Hauts-de-France. In the first experiment, the Acorn RISC Machine (ARM) embedded system was considered as a factory. The photovoltaic energy system was represented by the ARM embedded system with a luminosity sensor. The wind energy system was represented by the ARM embedded system with a temperature/humidity sensor. Figure 8 illustrates the implementation steps of the EasySched architecture composed of five ARM embedded systems.

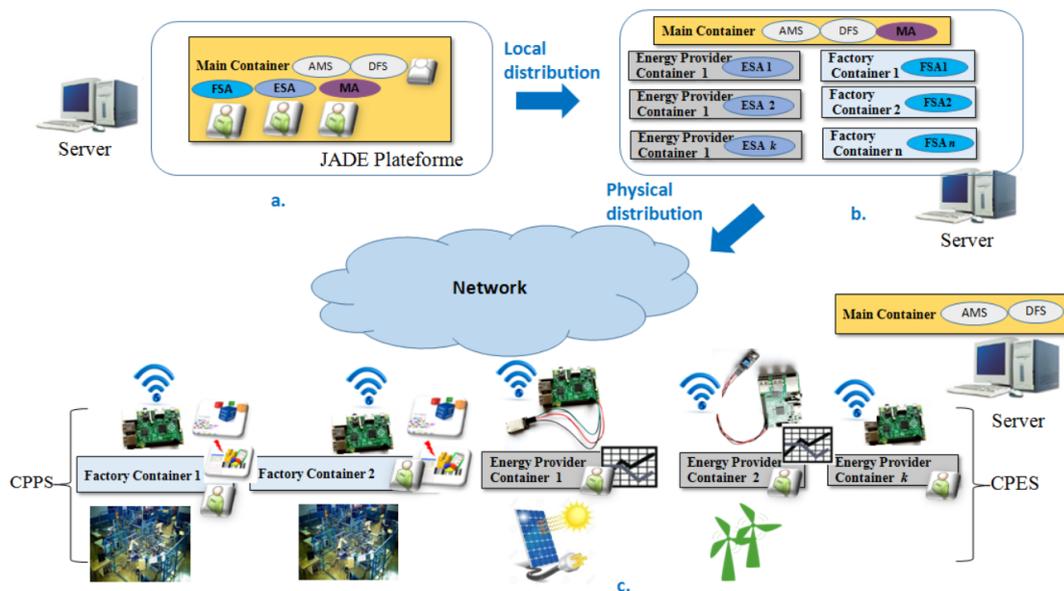


Figure 8. Implementation of EasySched: (a) Local distribution in one container, (b) Local distribution in different containers, (c) Physical distribution in different embedded networked containers.

The first step of the implementation was to propose agent behaviours of agents, negotiation, and coordinated mechanisms between agents and code them locally in the same main container (see Figure 8a). In a multi agent context, an agent requires the presence of a container. A container is an abstract class that contains all the services needed to host and manage agents during the execution of their behaviours (software program). These agent containers can be distributed throughout the network. The agents created are by default located in the same “Main container”. Before being deployed to the physically distributed system comprising connected embedded systems, a container is created for each distant agent (see Figure 8b). “Factory Container *i*” and “Energy Provider Container *i*” are thus containers allocated to agents FSA and ESA respectively (see Figure 8b). ARM A7 embedded systems are used to embed the ESA1, ESA2, FSA1, FSA2, and MA containers (see Figure 8c).

In order to validate the proposed dynamic mechanisms of the sustainable decision tools of CPPS and CPES, a set of perturbations is simulated. Luminosity, temperature, and humidity sensors are used to simulate the fluctuation of renewable energy. In our approach, the outputs of these sensors represented the variation in the quantities of certain parameters on which the power supplied by the two energy resources depends. For example, the variation in luminosity, which systematically influences the variation in the power delivered by any photovoltaic power generator is easily simulated by masking the light sensor to varying degrees. Similarly, the power delivered by a wind power generator depends on the temperature and humidity and can be modified by influencing the temperature and humidity sensors.

The pseudo code that allows the ESA to detect a fluctuation in energy provided by a wind generator is detailed in Algorithm 5. In the context of this work, the model proposed by (Marshall and Plumb., 2008) was used, where the electric power delivered by a wind turbine can be calculated as follows:

$$P = \frac{1}{2} \rho S V^3 \quad (3)$$

where S is the area of the wind turbine, V is the wind speed, and ρ is the air density.

The air density is calculated as follows in Equation (4):

$$\rho = \frac{1}{287.06(T + 273.15)} \left(p - 230.617 \times \phi \times \exp\left(\frac{17.5043 \times T}{241.2 + T}\right) \right) \quad (4)$$

where T and ϕ are the temperature and the humidity recovered from the sensors and the air pressure p equals 1013.25 hPa. Powers $P(t)$ and $P(0)$ are calculated using the previous equation. If the ratio $P(t)/P(0)$ is lower than 1, energy production decreases. The “Alarm” variable determining the system status is then updated.

Algorithm 5 Pseudo code of ESA Monitoring behaviour

- 1: Read data from sensors (temperature and humidity in our case)
 - 2: calculate the power $P(t)$
 - 3: **if** $P(t)/P(0) < 1$ **then**
 - 4: Alarm = “Perturbation Detected”
 - 5: **else**
 - 6: Alarm = “No perturbation”
 - 7: **end if**
-

From a technical point of view, to limit the effect of sensor noise on the signals and thus avoid considering ever change in signal as a disturbance, an artificial filtering algorithm was added as data pre-processing behaviour. For this, the period of this cyclic behaviour was bigger than the data acquisition behaviour period.

The pseudo code for this cyclic behavior is represented in Algorithm 6.

Algorithm 6 Pseudo code of ESA alarm triggering behaviour

- 1: **if** Alarm = “Perturbation detected” **then**
 - 2: timeResch = timeResch + $p2$
 - 3: Reschedule.timeResch = timeResch
 - 4: Reschedule.tauxEnergy = $(P(0) - P(t)/P(0)) \times 100$
 - 5: ControlMessage.Object = Reschedule
 - 6: Send (ControlMessage, FSA)
 - 7: $p2 = x \times p2$
 - 8: **else**
 - 9: ControlMessage.State = “No energy perturbation”
 - 10: Send(ControlMessage, FSA)
 - 11: **end if**
-

In order to avoid successive launches of rescheduling signals, the period $p2$ of cyclic behaviour is then modified when the ESA detects an anomaly (see line 7 in Algorithm 6).

5. Experimentation and Interpretation of Results

The distributed implementation consists of a PC and five ARM A7 embedded systems (FSA1, FSA2, ESA1, ESA2, and MA). The PC has an Intel Core i7 processor with 8 GB of RAM. The integrated card operates at 900 Mhz with 1 GB of RAM. The target embedded system runs the Debian version of Linux. All the machines run Java SE Embedded. A LAN is used for communication between all IoT components. The multi agent model was developed using the Java Agent Development Framework (JADE) multi agent platform in Netbeans IDE. To test the efficiency and the performance of the proposed EasySched architecture through its distributed implementation, different tests were performed on various types of job-shop benchmarks using machines with different speeds. Instances were named “machines_Vmax_Pi” where (machines) is the number of machines, (Vmax) is the maximum number of tasks per job and (Pi) is the range of processing times. For example, instance (3_5_10) refers to three machines with five operations per job and a processing time range of 10. Instance details are provided in [43]. Each instance was executed five times. For our experiments, the periods p_1 , p_2 , p_3 are initially set to 3 s, 6 s, and 8 s respectively. x is set to 3. Figure 9 presents a selection of real photos of our distributed implementation in the LAMIH Laboratory. The experiment comprises two stages: The first one concerns the validation of the predictive part of EasySched and the second concerns the assessment of the reactive part. Relevant performance measurements are provided in the Section 5.1.

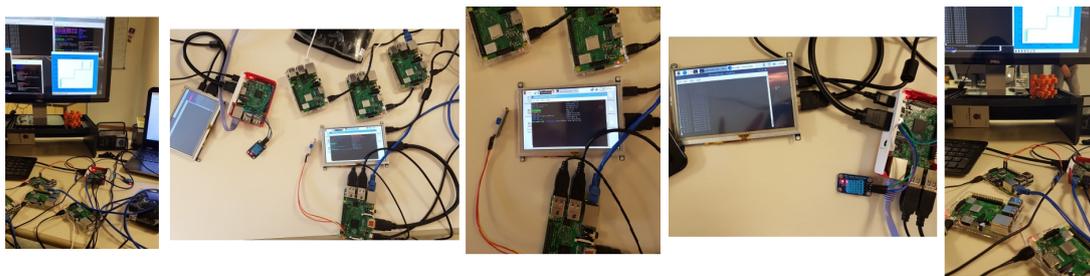


Figure 9. Implementation of EasySched.

5.1. Rescheduling Performance Measures

In order to assess the performance of the rescheduling method, performance indicators (KPI) had to be defined. In fact, in order to compare the quality of the initial schedule (predictive) and the reactive one (after rescheduling), it is necessary to define two types of performance measurements depending on the evolution of the effectiveness or the efficiency [42]. They are presented hereinafter.

- **Makespan Effectiveness:** Proposed in [56], it is defined as the percentage change in makespan of the predictive schedule compared to the original schedule.

$$ME = 1 - \frac{Makespan_{new} - Makespan_0}{Makespan_0} \times 100 \quad (5)$$

where $Makespan_{new}$ is the makespan of the repaired schedule using the new proposed rescheduling methods and $Makespan_0$ is the makespan of the original schedule. The maximum efficiency 100% is achieved if the makespan of the predictive and the reactive schedule are identical. If the increase in the makespan when a disruption occurs is minimal, the rescheduling process is efficient.

- **Energy Efficiency:** Is as a percentage change in energy consumed by the reactive schedule compared to the predictive one and is defined as follows:

$$EE = 1 - \frac{Energy_{new} - Energy_0}{Energy_0} \times 100 \quad (6)$$

where $Energy_{new}$ and $Energy_0$ represent respectively the reactive and the predictive energy consumption. Values greater than 100%, mean that the energy consumption of the reactive one is lower than the predictive one. This is due to the energy constraint threshold that varies in real time depending on the fluctuation in renewable energy.

5.2. Experimentation and Results of Predictive Part

In this subsection, the results of the predictive part are presented and discussed. As explained previously, the objective of the ESA in the CPES is to validate the predictive solutions found by the FSA. To assess the efficiency of the PSO algorithm used by the FSA, it is compared with the CPLEX CP optimizer results of Salido et al. [57]. CP optimizer was used as a reference to evaluate the efficiency and performance of our approach. Table 2 contains the results of the tests. The first column represents the names of the instances, following columns present the Makespan (Mk, in seconds) and the energy consumed (E in Wh) provided by the FSA1 and FSA2. The weighting parameter γ and the control message sent by the ESA are also specified. “Yes” refers to the approval of the predictive schedules, whereas “No” triggers the FSA to find another solution after negotiating with other FSAs while reducing the weighting parameter γ . The last columns represents Mk and E found by the CP Optimizer in [57]. As it can be seen, the PSO results are efficient in terms of the quality of the solutions found. The Mk values are the same for small instances compared with the CP Optimizer. The Mk and E are very close to the CP optimizer values. The total average deviation between the results of FSA1 and the CP Optimizer is 2.8% for makespan and 0.6% for energy consumption. More details are described in Appendix A.

Table 2. Results of the predictive part.

Instance	FSA 1				FSA 2				CP Optimizer	
	γ	Mk	E	Message	γ	Mk	E	Message	Mk	E
3_5_10	1	41	145.8	No	1	41	145.8	No	41	143.1
	0.9	41	133.7	No	0.9	41	133.2	Yes	-	-
	0.8	42.2	123.5	No	0.8	-	-	-	-	-
	0.7	45.3	112.6	Yes	0.7	-	-	-	-	-
7_10_100	1	625.9	2773.4	No	1	625.9	2773.4	No	625.9	2664.1
	0.9	626	2560.7	Yes	0.9	626	2560.7	No	-	-
	-	-	-	-	0.8	642.4	2418.9	Yes	-	-
3_25_100	1	1711.8	6797.2	No	1	1711.8	6797.2	No	1673.4	6732.2
	0.9	1732.2	6311	No	0.9	1732.2	6311	No	-	-
	0.8	1791.4	5726.2	No	0.8	1791.4	5726.2	No	-	-
	0.7	1943.7	5097.4	No	0.7	1886.3	6107.7	No	-	-
	0.6	2118.3	4617.4	Yes	0.6	2197.6	4547.9	No	-	-
50_20_100	0.5	-	-	-	0.5	2317.1	4257.3	Yes	-	-
	1	5396.8	78,976.6	No	0.7	5559.8	75,243.6	No	5008.3	81,527.7
	0.6	5657.8	73,653.8	Yes	0.6	5246.1	70,342	Yes	-	-
100_20_100	0.8	9785.8	150,468.8	No	0.6	8761.3	145,892.3	No	9434.5	152,047.9
	0.5	9769.5	147,480	Yes	0.5	9769.5	14,956.4	Yes	-	-

These predictive experiments also aim to validate the negotiation protocol between agents (FSA and ESA) presented in Algorithm 3. The search for the predictive solution is repeated until the approval message “yes” is received from the ESA indicating that the CPES has sufficient energy to satisfy demands.

As one can see from Table 2, the ESA sends approval messages only if the sum of the energy demand is lower than or equals to a fixed predictive energy threshold. For instance 1, the ESA sends a “Yes” message when the energy is lower than 5000 in instance 2, etc. These results validate the effectiveness of the negotiation protocol between factory schedulers and energy schedulers.

Figure 10 illustrates the predictive solution delivered by FSA 1 after executing the PSO algorithm. The makespan of this solution equals 44 s and the energy consumption is 133 Wh. The particle (the PSO algorithm output) corresponding to this solution and the Gantt chart are given in Figure 10. The machine availability vector in the particle representation is useful to extract the affected operations in case of disruption and to follow its propagation (see Figure 10). Figure 11 illustrates the profiling of machine speed used in the predictive solution. As we can see, in this solution the machines are used in high speed to favor the reduction of the completion time. Figure 12 illustrates the profiling of the energy consumption per machine with the predictive solution. The idle energy consumption, that refers to the consumption during the slack times, is not calculated in this work because the power of the machines is not specified in the instances in [43]. However, it remains a very important characteristic that affects the total energy consumption.

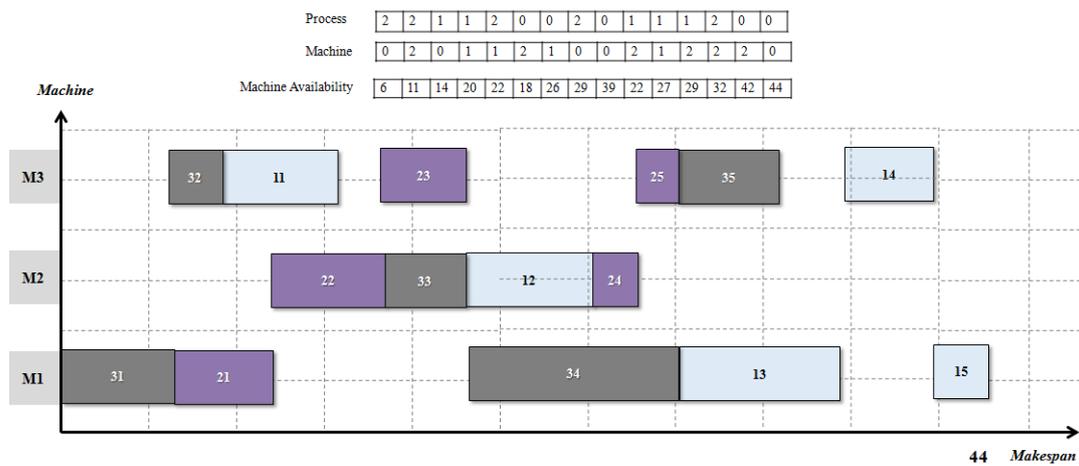


Figure 10. A predictive solution found by FSA 1.

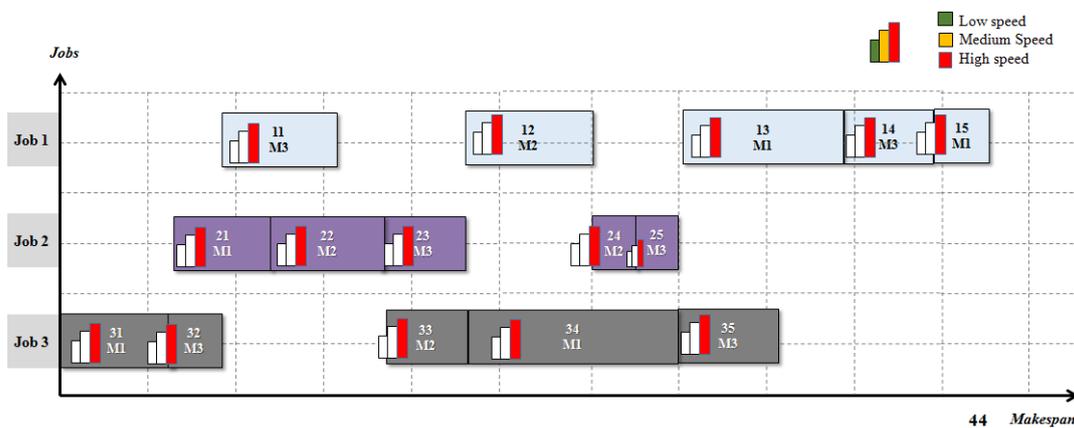


Figure 11. Profiling of the machine speed used in the predictive solution.

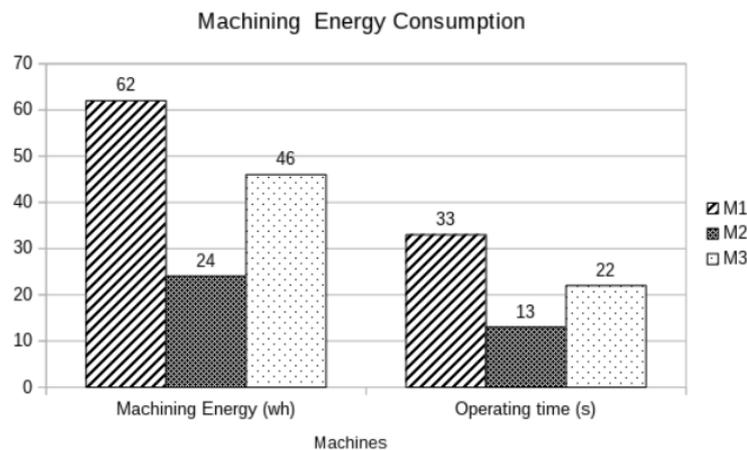


Figure 12. Profiling of the machine energy consumption of the predictive solution.

5.3. Experimentation and Results of the Reactive Part

The objective of the experimentation is to validate the reactive part of the proposed architecture. For this purpose, many perturbations were simulated to affect the predictive solutions. As mentioned before, the fluctuation of the renewable energy is simulated by varying the output of the luminosity, temperature and humidity sensors. Figure 13 illustrates the sensing data results acquired by the ESA after executing the monitoring behaviour. The analysis of the data enables any fluctuations in renewable energy to be detected. As in the predictive part the makespan (Mk) and the energy consumption (EC), which reflect production and energy efficiency respectively are estimated.

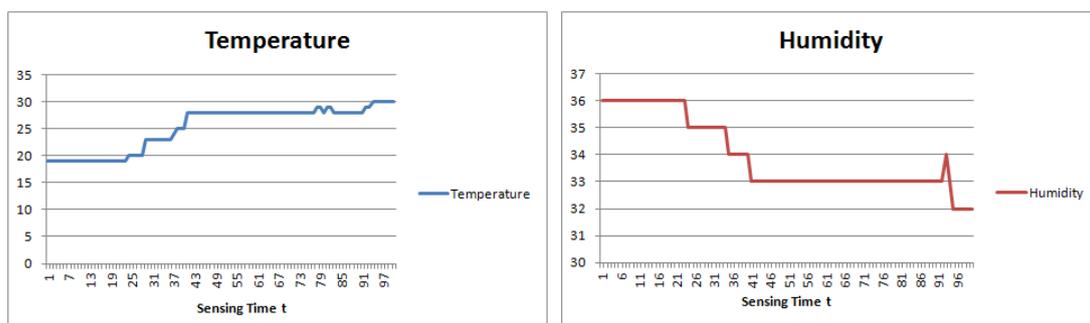


Figure 13. Data acquired by the ESA when executing the monitoring behaviour.

Table 3 summarizes the results of the reactive experiments when faced with energy perturbations. The makespan and the energy consumption of predictive and reactive solutions are compared. Table 3 contains the Mk and EC of the predictive and reactive solutions respectively while specifying the energy threshold communicated by the ESA that controls this factory. It compares the performance of the reactive solutions found by the rescheduling methods with the predictive solutions. As one can see, in most cases, the FSA was able to find a new solution respecting the energy constraint.

Figure 14 contains the Gantt diagram of a reactive schedule found by the factory scheduler agent FSA1 after executing the rescheduling method. As we can see from this figure, a new energy threshold was detected and communicated by the CPES. This trigger launches the rescheduling procedure at time 22 s (trigger time). The energy consumption of the reactive schedule found was reduced to 94 Wh after the disruption while the makespan is degraded to 53 s (see Figure 14). As we can see, the energy consumption and the makespan of the predictive and the reactive schedules are not the same. However, there is a small deviation in Mk value from 44 s to 53 s and in EC value from 133 Wh to 94 Wh. The new energy consumption of the reactive schedule respected the energy threshold 26%. Figure 15 illustrates the variation of the speed of machines of the affected operations to reduce the

energy consumption. The high machine speeds were reduced to low and medium speed. In this scenario, the adaptation of the energy consumption of the factory to respect the 26% energy threshold led to a deterioration in the makespan performance (an increase in the production completion time). However, the quantification of this degradation depends on the production and energy consumption models used. This solution was found using the first rescheduling method proposed that seeks to preserve stability and change the speeds of machines to meet the energy constraint. The speeds of the machines M1, M2, and M3 are thus changed starting from the rescheduling time (see Figure 15).

Table 3. Results of the reactive part.

Instance	FSA 1			ESA 1	FSA 2			ESA 2
	Predictive (Mk, E)	Mk	E	Energy Threshold %	Predictive (Mk, E)	Mk	E	Energy Threshold %
3_5_10	44, 133	53	94	26	626, 2773	42	123	10
7_10_100	626, 2773	703	2099	22	626, 2773	642	2418	10
3_25_100	1711, 6797	2056	4742	28	1711, 679	1854	5466	10
50_20_100	5396.8, 78,976.6	8359	61,635.3	22	5559.8, 75,243.6	5305.1	70,358	10
100_20_100	9785.8, 150,468.8	8761.3	145,892.3	23	9769.5, 147,480	11048	132,690.7	10

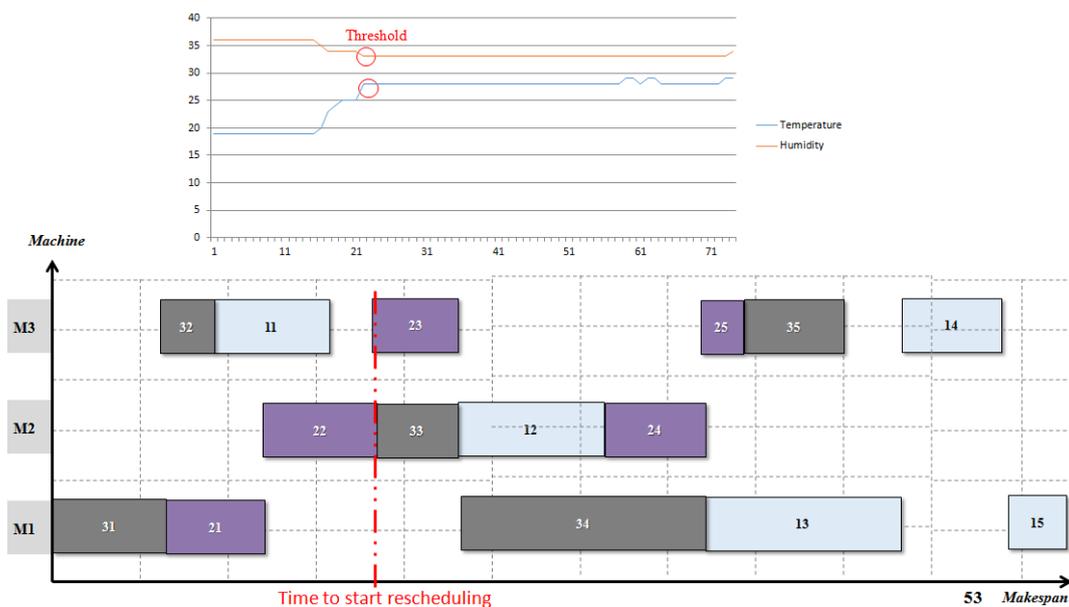


Figure 14. The reactive solution found by EasySched.

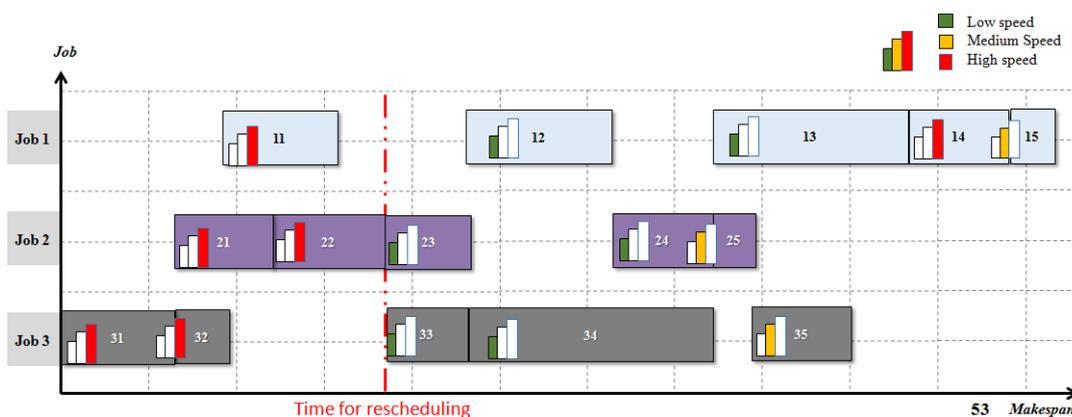


Figure 15. Profiling of the machine speed used in the reactive solution.

Figures 16 and 17 show the deviation between the predictive and reactive makespan and energy consumption respectively. The results obtained validate the efficiency and the reactivity of the proposed multi agent architecture.

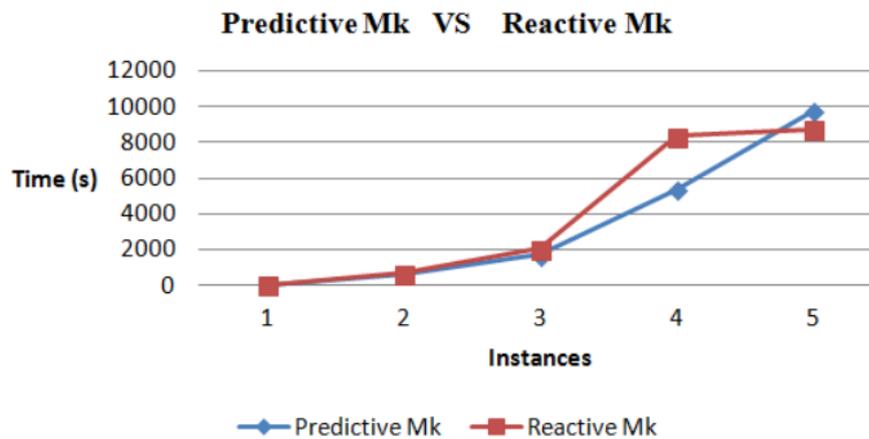


Figure 16. Makespan (Mk) found by EasySched.

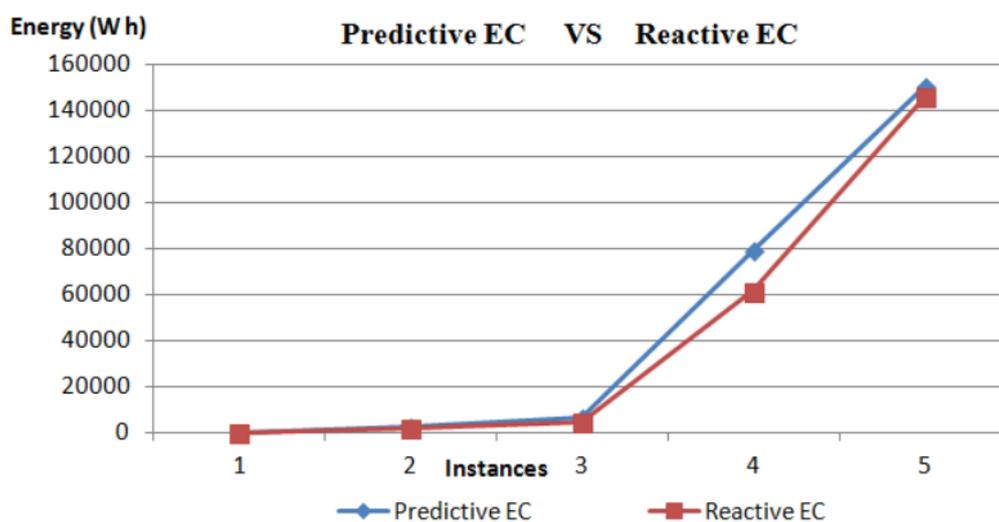


Figure 17. Energy consumption (EC) found by EasySched.

Table 4 presents the makespan efficiency and the energy efficiency for the different instances. For instance 1, Makespan Efficiency (ME) equals 79%, which means that the makespan has not degraded much compared to the predictive makespan. However, Energy Efficiency (EE) equals 129%, which means a decrease in EC with regards to the energy threshold. Thus, the reactive energy value is lower than the predictive one.

Table 4. Makespan and energy efficiency of the proposed method.

Instance	Makespan Efficiency	Energy Efficiency
3_5_10	79.5454	129.323
7_10_100	87.6996	124.305
3_25_100	79.836	130.233
50_20_100	45.1119	121.957
100_20_100	110.469	103.041

Even though the results seem to be in favour of the proposed EasySched, the simulation results obtained for all the cases were examined using an analysis of variance (one way ANOVA) to conclude if there was a statistically significant difference to support our conclusions. A statistical sensitivity test has been performed on the results.

In order to validate the obtained results, an ANOVA analysis is performed. The results obtained with the one-way ANOVA are presented in Table 5. The F-ratio and the p -value show the impact of the perturbation on the quality of the predictive solution.

The impact is considered significant for p -values < 0.05 . However, according to the p -values in Table 5, the perturbation does not have a significant impact.

Table 5. One-way analysis of variance (ANOVA) comparing Mk and EC of the proposed method.

ANOVA	Makespan	Energy Consumption
p -value	0.861	0.906
F-ratio	0.032	0.014

The results presented above represent the first implementation case where each CPPS was connected to a different CPES (factory 1 to wind generator, factory 2 to photovoltaic generator). In the next subsection, the results of the second situation are analysed.

5.4. Self Configuration of CPES

Contrary to the configuration in the previous section, the results presented in this section were obtained from the second implementation case of EasySched where all CPPSs were connected to the same CPES. Tables 6 and 7 summarize the results of the experiments related to the dynamic configuration of the weighting parameter to switch between energy sources. Some assumptions are taken into consideration. Firstly, all renewable energy generators (represented by CPES) were combined and controlled by a global ESA delegated to make decisions to control the distribution of energy portion from each generator autonomously. As the luminosity sensor used in our experiments was digital and not analogue, the output was binary (0 or 1) and therefore, we could not quantify the power delivered by the photovoltaic generator. However, in this case, the weighting parameter β was set to 0.5 when the output (sensor data) was 1 or 0.2 otherwise. If there is a decrease in wind energy, α is decreased. If there is a power degradation by 10% for example α is reduced by 0.1, for 20% α is reduced by 0.2, etc.

Table 6. Self-configuration of weighting parameter according to the luminosity sensor output.

Output of Luminosity Sensor	α	β
1	0.5	0.5
0	0.8	0.2

Table 7. Self-configuration of weighting parameter according to degradation of the wind power.

Degradation of Wind Power %	α	β
10	0.8	0.2
20	0.7	0.3
40	0.5	0.5

As we can see from this experiment, the ESA was able to provide internal intelligent decisions to self-configure the weighting parameter responsible for switching the energy supply depending on the fluctuation in the power of each renewable energy generator.

Other scenarios can be imagined in the context of switching between energy sources, for example, the dynamical real-time variation in the price could also be treated as a source of perturbation.

5.5. Discussion

The results obtained by EasySched architecture seem to be promising in terms of makespan and energy efficiency of the obtained solutions. The architecture enables the connected factories to adjust especially their consumption in terms of an energy threshold communicated in real-time by CPES. A statistical sensitivity test has been performed on the results. However, formal studies have to be conducted to prove the effectiveness and efficiency in predictive and reactive parts. Other self configuration mechanisms have to be designed for the CPES to switch between different renewable sources.

6. Conclusions

This work deals with sustainable manufacturing systems to improve resource efficiency, environmental friendliness, and energy-aware production performance. A multi-agent architecture named “EasySched” was proposed to find predictive and reactive productionscheduling based on renewable energy that is difficult to predict. The agents proposed in “EasySched” are responsible for the smart sustainable decision tools of the cyber part of cyber physical production systems and cyber physical energy systems. The smart sustainable tool of CPPS and CPES is an agent based model coupled with intelligent techniques (PSO) and dynamic interactions. The proposed architecture contains intelligent mechanisms for enhancing adaptability to events. The objective of the proposed mechanisms was to coordinate the communication between CPPS and CPES in order to adjust the consumption of factories in terms of an energy threshold communicated in real time by CPES. To assess the performance of the predictive and reactive parts, several instances resulting from the literature were tested on an original physical distributed implementation of the architecture. The results illustrate only the efficiency and the effectiveness of the proposed approach in specific case studies for both the predictive and reactive parts. This means that formal studies have to be conducted to prove the contribution of our work. Other experiments on the capabilities of CPES to self configure and switch between different sources of renewable energy will be the object of future research. A second aspect is to provide a multi-layer holonic architecture to generalize EasySched. In this case, the scheduler representing the connected factory would be a holon recursively composed of a set of holons each representing a machine or intelligent product in the industry. A third perspective is to add a prediction module to the sustainable decision tool of CPS, based on artificial intelligence techniques such as recurrent neural networks (RNN), to solve one of the other main challenges in this area which is the computational prediction of system uncertainty.

Author Contributions: Conceptualization of the model was done by M.N. and D.T. Simulations, validation of results and writing tasks of this research work was carried out by M.N. Supervision of the work, interpretation, investigation of the results and editing of the manuscript were done by D.T., M.N. and A.B.

Funding: This research was funded by the ELSAT2020 project of CPER sponsored by the French ministry of sciences, the Hauts de France region and the European Regional Development Fund (ERDF/FEDER).

Acknowledgments: The ELSAT2020 project is co-financed by the European Union with the European Regional Development Fund, the French state and the Hauts de France Region Council. The authors are grateful to these institution.

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

Abbreviations

The following abbreviations are used in this manuscript:

CHS	Cyber Human System
CPES	Cyber Physical Energy System
CPPS	Cyber Physical Production System
CPS	Cyber Physical System
EAS	Energy-Aware Scheduling

EasySched	Energy-Aware production SYstem SCHEDuling
EC	Energy Consumption
EE	Energy Efficiency
EER	Energy-Efficient Rescheduling
EES	Energy-Efficient Scheduling
EMS	Energy Management System
ERP	Entreprise Resource Planning
ES	Energy-efficient Scheduling
ESA	Energy Scheduler Agent
JSP	Job Shop Problem
JSMS	Job Shop with Machines at different speeds
FJSP	Flexible Job Shop Problem
NSGA	Non-dominated Sorting Genetic Algorithm
CP	Constraint Programming
FSA	Factory Scheduler Agent
ESA	Energy Scheduler Agent
DS	(Directory Facilitator)
DFS	Distributed File System
LAN	Local Area Network
AMS	Agent Management System
JADE	Java Agent Development Framework
RFID	Radio Frequency Identification
ARM	Acorn RISC Machine
FMS	Flexible Manufacturing System
FMS	Flexible Manufacturing System
FSA	Factory Scheduler Agent
IoT	Internet of Things
ME	Makespan Efficiency
MAS	Multi Agent System
MILP	Mixed Integer Linear Programming
PSO	Particle Swarm Optimization
SIMS	Sustainable Intelligent Manufacturing System

Appendix A

Appendix A.1. Job Shop Scheduling Problem with Machines at Different Speeds

Job shop scheduling problem with machine at different speeds (JSMS) is an extension from classical job shop problem (JSP). JSP can be formulated by a set of jobs ($j = 1 \dots J$) with a set of machines ($m = 1 \dots M$). In JSP, each job has a set of operation ($i = 1 \dots I$). Each operation O_{ij} requires a duration p_{ij} . Each operation O_{ij} has a machine to be executed on. The problem consists of sequencing the operations on machines in order to optimize a given objective while respecting a set of given constraints.

These constraints define the limits that should be considered when solving the problem. In JSP each machine can process only one operation at time. Once the execution of an operation starts, the pre-emption is not allowed. Precedence constraints between operations of the same job should be respected. There are no precedence relationships between operations of different jobs. The objective to reach is time-based. It is often expressed in terms of total production time (leadtime) or sometimes, in terms of just-in-time production.

In JSMS, each machine can work with different speeds or modes. An operation of one job can be executed on a machine at a given speed to be chosen. The speed of machine affects the processing time and the energy consumed during the execution of the operation. The processing time decreases when the machine works with higher speed while the energy consumption increases. Each mode or speed of machine has its corresponding processing and energy consumption. Solving the JSMS problem refers

to determining the sequencing of operations on machines and their speeds that minimize a criterion based on the makespan and the total energy consumption. The speed of a machine cannot be changed during the execution of an operation.

Appendix A.2. Constraint Programming Model of JSMS

Constraint programming (CP) is a modeling language. It is a powerful domain filtering algorithm for global constraints and very effective on complex scheduling problems and other hard combinatorial optimization problems [58].

The JSP problem is modelled as a constraint satisfaction problem (CSP) with a triple (X, D, C) where X is a set of variables, D is a mapping of variables to finite sets of integer values, where $D(x)$ representing the domain of x . In constraint programming model, each CP variable is associated to an interval. The CP Optimizer model uses many constraints such as “Logic constraint”, etc. The process of removing inconsistent values from the variables domain is called propagation. In JSP, each task O_{ij} has a single machine assignment M_{ijr} and a start time St_{ijr} to be determined. In JSMS, each machine can work with different speeds, the duration and energy combination is presented by a tuple p_{ij}, e_{ij} . The objective function is to find a schedule that reduces in an aggregated and normalized way the makespan and the energy consumption. The Algorithm A1 shows the pseudo code of the CP Model proposed in [59] to solve the JSMS problem.

Algorithm A1 Pseudo code of CP-Optimizer Model

- 1: **Data:** *tasks*: set of tasks; *modes*: Set of 3 modes for each task;
 - 2: **Result:** A solution minimizing the objective function depending of γ
 - 3: *interval itvs* := Define interval, one for each task;
 - 4: *interval modes* := Define mode, three for each task;
 - 5: *Sequence mchs* := Each *itvs* with the same machine is linked up;
 - 6: **Minimize** $\gamma \times NormMakespan + (1 - \gamma) \times NormEnergy$
 - 7: **Subject to**
 - 8: noOverlap (*machs*)
 - 9: endBeforeStart (*itvs[j][o]*, *itvs[j][o + 1]*)
 - 10: alternative (*itvs[j][o]*, all (*md* in *modes*; if (*md.id*==*itvs.id*)))
-

NormEnergy and *NormMakespan* refer to the normalization of both measures, where $NormMakespan = \frac{makespan}{maxMakespan}$ and $NormEnergy = \frac{Energy}{maxEnergy}$.

MaxMakespan is the sum of operation durations when all machines are working with the lowest speed. *MaxEnergy* is the sum of energy consumed in the execution of operations when all machines are working with the highest speed. For more details about CP optimizer of JSMS, one can see [59,60]. For JSMS mathematical model, one can see [61], but this is not our concern in this paper since the authors did not consider the same benchmark, objectives, etc.

Appendix A.3. Instances

The proposed model was tested on different instances that varies according to the number of jobs operations, machines, etc. Instances are named “machines_Vmax_Pi” where (machines) is the number of machines, (Vmax) is the maximum number of operations per job and (P_i) is the range of processing times.

The tables below present the details of a small instance (3_5_10) with three machines, five operations per job, and a range of processing time set to 10 time units. All machines of all instances have three modes or speeds. Table A1 presents the machine assignment of operations of each job. For each operation of each job, the $\langle id, m_{ijr} \rangle$ is given. $\langle 1, 2 \rangle$ means the first operation of job 1 is assigned to machine 2. For each operation, three triplets are represented: the *id* of the task, the energy used e_{ij} and the processing time p_{ij} ($\langle id, e_{ij}, p_{ij} \rangle$). Table A2 presents the processing time of each

operation depending on the speed of the machine. Table A3 presents the energy consumption of each operation depending on the speed of the machine. The data of all tested instances are in [57,62].

Table A1. Machine assignment of operations of the small instance.

	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5
Job 1	<1,2>	<2,1>	<3,0>	<4,2>	<5,0>
Job 2	<6,0>	<7,1>	<8,2>	<9,1>	<10,2>
Job 3	<11,0>	<12,2>	<13,1>	<14,0>	<15,2>

Table A2. Processing time of operations of the small instance.

	Speed 1	Speed 2	Speed 3
	14	10	7
	4	6	5
	11	15	10
	5	4	3
	4	2	3
	12	14	8
	6	9	7
	4	3	2
	1	3	2
	2	3	1
	6	8	9
	10	7	5
	2	4	3
	10	9	7
	4	3	6

Table A3. Energy consumption of operations of the small instance.

	Speed 1	Speed 2	Speed 3
	14	16	19
	6	4	5
	14	13	15
	3	4	5
	1	2	3
	17	16	18
	12	5	10
	2	3	4
	2	1	1
	1	1	2
	17	12	8
	4	7	13
	2	1	1
	3	5	9
	5	6	3

References

- Schmidt, R.; Möhring, M.; Härting, R.C.; Reichstein, C.; Neumaier, P.; Jozinović, P. Industry 4.0-potentials for creating smart products: Empirical research results. In *International Conference on Business Information Systems*; Springer: Berlin, Germany, 2015; pp. 16–27.
- Farhangi, H. The path of the smart grid. *IEEE Power Energy Mag.* **2010**, *8*, 18–28. [[CrossRef](#)]
- Ilic, M.D.; Xie, L.; Khan, U.A.; Moura, J.M. Modeling of future cyber–physical energy systems for distributed sensing and control. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2010**, *40*, 825–838. [[CrossRef](#)]
- Frazzon, E.M.; Dutra, M.L.; Vianna, W.B. Big data applied to cyber-physical logistic systems: Conceptual model and perspectives. *Braz. J. Op. Prod. Manag.* **2015**, *12*, 330–337. [[CrossRef](#)]

5. Zhong, R.Y.; Xu, X.; Klotz, E.; Newman, S.T. Intelligent manufacturing in the context of industry 4.0: A review. *Engineering* **2017**, *3*, 616–630. [[CrossRef](#)]
6. Lee, J.; Bagheri, B.; Kao, H.A. A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufact. Lett.* **2015**, *3*, 18–23. [[CrossRef](#)]
7. Zanni, A. Cyber-physical systems and smart cities. *IBM Big Data Anal.* **2015**, *20*. Available online: <https://www.semanticscholar.org/paper/Cyber-physical-systems-and-smart-cities-Learn-how-%2C-Zanni/20d92085910dab766fcf6e5897f01d734db7c23d> (accessed on 16 November 2019).
8. Petrolo, R.; Loscri, V.; Mitton, N. Towards a smart city based on cloud of things, a survey on the smart city vision and paradigms. *Trans. Emerg. Telecommun. Technol.* **2017**, *28*, e2931. [[CrossRef](#)]
9. Thoben, K.D.; Wiesner, S.; Wuest, T. “Industrie 4.0” and smart manufacturing—a review of research issues and application examples. *Int. J. Autom. Technol.* **2017**, *11*, 4–16. [[CrossRef](#)]
10. Konstantinos, I.; Georgios, T.; Garyfalos, A. A Decision Support System methodology for selecting wind farm installation locations using AHP and TOPSIS: Case study in Eastern Macedonia and Thrace region, Greece. *Energy Policy* **2019**, *132*, 232–246. [[CrossRef](#)]
11. Ioannou, K.; Tsantopoulos, G.; Arabatzis, G.; Andreopoulou, Z.; Zafeiriou, E. A spatial decision support system framework for the evaluation of biomass energy production locations: Case study in the regional unit of drama, Greece. *Sustainability* **2018**, *10*, 531. [[CrossRef](#)]
12. Prabhu, V.V.; Trentesaux, D.; Taisch, M. Energy-aware manufacturing operations. *Int. J. Prod. Res.* **2015**, *53*, 6994–7004. [[CrossRef](#)]
13. Pach, C.; Berger, T.; Sallez, Y.; Trentesaux, D. Reactive control of overall power consumption in flexible manufacturing systems scheduling: A Potential Fields model. *Control Eng. Pract.* **2015**, *44*, 193–208, doi:10.1016/j.conengprac.2015.08.003. [[CrossRef](#)]
14. Feng, Y.; Wang, Q.; Gao, Y.; Cheng, J.; Tan, J. Energy-Efficient Job-Shop Dynamic Scheduling System Based on the Cyber-Physical Energy-Monitoring System. *IEEE Access* **2018**, *6*, 52238–52247. [[CrossRef](#)]
15. Liao, Y.; Deschamps, F.; Loures, E.d.F.R.; Ramos, L.F.P. Past, present and future of Industry 4.0—a systematic literature review and research agenda proposal. *Int. J. Prod. Res.* **2017**, *55*, 3609–3629. [[CrossRef](#)]
16. Monostori, L. Cyber-physical production systems: Roots, expectations and R&D challenges. *Proc. Cirp* **2014**, *17*, 9–13.
17. Wang, W.; Hong, T.; Li, N.; Wang, R.Q.; Chen, J. Linking energy-cyber-physical systems with occupancy prediction and interpretation through WiFi probe-based ensemble classification. *Appl. Energy* **2019**, *236*, 55–69. [[CrossRef](#)]
18. Steinbrink, C.; Blank-Babazadeh, M.; El-Ama, A.; Holly, S.; Lüers, B.; Nebel-Wenner, M.; Ramírez Acosta, R.P.; Raub, T.; Schwarz, J.S.; Stark, S. CPES Testing with mosaik: Co-Simulation Planning, Execution and Analysis. *Appl. Sci.* **2019**, *9*, 923. [[CrossRef](#)]
19. Cardin, O. Classification of cyber-physical production systems applications: Proposition of an analysis framework. *Comput. Ind.* **2019**, *104*, 11–21. [[CrossRef](#)]
20. Liu, C.; Jiang, P. A cyber-physical system architecture in shop floor for intelligent manufacturing. *Proc. Cirp* **2016**, *56*, 372–377. [[CrossRef](#)]
21. Keil, S. Design of a cyber-physical production system for semiconductor manufacturing. In *Proceedings of the Hamburg International Conference of Logistics (HICL)*; Hamburg University of Technology: Hamburg, Germany, 2017; pp. 319–340. Available online: <https://pdfs.semanticscholar.org/807e/73da33fa56f5467c0bb0319d323c352630c5.pdf> (accessed on 16 November 2019).
22. Macana, C.A.; Quijano, N.; Mojica-Nava, E. A Survey on Cyber Physical Energy Systems and their Applications on Smart Grids. In *Proceedings of the 2011 IEEE PES Conference on Innovative Smart Grid Technologies Latin America (ISGT LA)*, Medellin, Colombia, 19–21 October 2011; pp. 1–7.
23. Zhang, F.; Shi, Z.; Wolf, W. A dynamic battery model for co-design in cyber-physical systems. In *Proceedings of the 2009 29th IEEE International Conference on Distributed Computing Systems Workshops*, Montreal, QC, Canada, 22–26 June 2009; pp. 51–56.
24. Yang, Q. Internet of things application in smart grid: A brief overview of challenges, opportunities, and future trends. In *Smart Power Distribution Systems*; Elsevier: Amsterdam, The Netherlands, 2019; pp. 267–283.
25. Shi, Z.; Yao, N.; Zhang, F. Scheduling feasibility of energy management in micro-grids based on significant moment analysis. In *Cyber-Physical Systems*; Elsevier: Amsterdam, The Netherlands, 2017; pp. 431–449.

26. Li, Z.; Huang, P.C.; Mok, A.K.; Nghiem, T.; Behl, M.; Pappas, G.; Mangharam, R. On the feasibility of linear discrete-time systems of the green scheduling problem. In Proceedings of the 2011 IEEE 32nd Real-Time Systems Symposium, Vienna, Austria, 29 November–2 December 2011; pp. 295–304.
27. Wu, W.; Aziz, M.K.; Huang, H.; Yu, H.; Gooi, H.B. A real-time cyber-physical energy management system for smart houses. In Proceedings of the 2011 IEEE PES Innovative Smart Grid Technologies, Perth, WA, Australia, 13–16 November 2011; pp. 1–8.
28. Mouzon, G.; Yildirim, M.B. A framework to minimise total energy consumption and total tardiness on a single machine. *Int. J. Sustain. Eng.* **2008**, *1*, 105–116. [[CrossRef](#)]
29. Fang, K.; Uhan, N.; Zhao, F.; Sutherland, J.W. A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction. *J. Manuf. Syst.* **2011**, *30*, 234–240. [[CrossRef](#)]
30. He, Y.; Li, Y.; Wu, T.; Sutherland, J.W. An energy-responsive optimization method for machine tool selection and operation sequence in flexible machining job shops. *J. Clean. Prod.* **2015**, *87*, 245–254. [[CrossRef](#)]
31. Tonelli, F.; Bruzzone, A.; Paolucci, M.; Carpanzano, E.; Nicolò, G.; Giret, A.; Salido, M.; Trentesaux, D. Assessment of mathematical programming and agent-based modelling for off-line scheduling: Application to energy aware manufacturing. *CIRP Ann.* **2016**, *65*, 405–408. [[CrossRef](#)]
32. Jiang, S.L.; Zhang, L. Energy-oriented scheduling for hybrid flow shop with limited buffers through efficient multi-objective optimization. *IEEE Access* **2019**, *7*, 34477–3487. [[CrossRef](#)]
33. Liu, G.S.; Yang, H.D.; Cheng, M.B. A three-stage decomposition approach for energy-aware scheduling with processing-time-dependent product quality. *Int. J. Prod. Res.* **2017**, *55*, 3073–3091. [[CrossRef](#)]
34. Zhang, Z.; Wu, L.; Peng, T.; Jia, S. An Improved Scheduling Approach for Minimizing Total Energy Consumption and Makespan in a Flexible Job Shop Environment. *Sustainability* **2019**, *11*, 179. [[CrossRef](#)]
35. Li, H.; Cao, H. An Optimization Model for Carbon Efficiency of a Job-shop Manufacturing System. *Proc. CIRP* **2015**, *28*, 113–118. [[CrossRef](#)]
36. Piroozfard, H.; Wong, K.Y.; Wong, W.P. Minimizing total carbon footprint and total late work criterion in flexible job shop scheduling by using an improved multi-objective genetic algorithm. *Resour. Conserv. Recycl.* **2018**, *128*, 267–283. [[CrossRef](#)]
37. Moon, J.Y.; Park, J. Smart production scheduling with time-dependent and machine-dependent electricity cost by considering distributed energy resources and energy storage. *Int. J. Prod. Res.* **2014**, *52*, 3922–3939. [[CrossRef](#)]
38. Mikhaylidi, Y.; Naseraldin, H.; Yedidsion, L. Operations scheduling under electricity time-varying prices. *Int. J. Prod. Res.* **2015**, *53*, 7136–7157. [[CrossRef](#)]
39. Liu, C.; Cai, W.; Jia, S.; Zhang, M.; Guo, H.; Hu, L.; Jiang, Z. Energy-based evaluation and improvement for sustainable manufacturing systems considering resource efficiency and environment performance. *Energy Convers. Manag.* **2018**, *177*, 176–189. [[CrossRef](#)]
40. Chaari, T.; Chaabane, S.; Aissani, N.; Trentesaux, D. Scheduling under uncertainty: Survey and research directions. In Proceedings of the 2014 IEEE International Conference on Advanced Logistics and Transport (ICALT), Hammamet, Tunisia, 1–3 May 2014; pp. 229–234.
41. Qiao, F.; Ma, Y.; Zhou, M.; Wu, Q. A novel rescheduling method for dynamic semiconductor manufacturing systems. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, 1–11. [[CrossRef](#)]
42. Nouiri, M.; Bekrar, A.; Trentesaux, D. Towards energy efficient scheduling and rescheduling for dynamic flexible job shop problem. *IFAC-PapersOnLine* **2018**, *51*, 1275–1280. [[CrossRef](#)]
43. Salido, M.A.; Escamilla, J.; Barber, F.; Giret, A. Rescheduling in job-shop problems for sustainable manufacturing systems. *J. Clean. Prod.* **2017**, *162*, S121–S132. [[CrossRef](#)]
44. Zhang, L.; Li, X.; Gao, L.; Zhang, G. Dynamic rescheduling in FMS that is simultaneously considering energy consumption and schedule efficiency. *Int. J. Adv. Manuf. Technol.* **2016**, *87*, 1387–1399. [[CrossRef](#)]
45. Nouiri, M.; Bekrar, A.; Jemai, A.; Ammari, A.C.; Niar, S. A new rescheduling heuristic for flexible job shop problem with machine disruption. In *Service Orientation in Holonic and Multi-Agent Manufacturing*; Springer: Berlin, Germany, 2018; pp. 461–476.
46. 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](#)]
47. Trentesaux, D.; Giret, A.; Tonelli, F.; Skobelev, P. Emerging key requirements for future energy-aware production scheduling systems: A multi-agent and holonic perspective. In *International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing*; Springer: Berlin, Germany, 2016; pp. 127–141.

48. Nouiri, M.; Trentesaux, D.; Bekrar, A.; Giret, A.; Salido, M.A. Cooperation Between Smart Manufacturing Scheduling Systems and Energy Providers: A Multi-agent Perspective. In *International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing*; Springer: Berlin, Germany, 2018; pp. 197–210.
49. Li, C.; Qouneh, A.; Li, T. iSwitch: Coordinating and optimizing renewable energy powered server clusters. *ACM SIGARCH Comput. Archit. News* **2012**, *40*, 512–523. [[CrossRef](#)]
50. Riveros, J.Z.; Bruninx, K.; Poncelet, K.; D'haeseleer, W. Bidding strategies for virtual power plants considering CHPs and intermittent renewables. *Energy Convers. Manag.* **2015**, *103*, 408–418. [[CrossRef](#)]
51. May, G.; Barletta, I.; Stahl, B.; Taisch, M. Energy management in production: A novel method to develop key performance indicators for improving energy efficiency. *Appl. Energy* **2015**, *149*, 46–61. [[CrossRef](#)]
52. May, G.; Stahl, B.; Taisch, M. Energy management in manufacturing: Toward eco-factories of the future—A focus group study. *Appl. Energy* **2016**, *164*, 628–638. [[CrossRef](#)]
53. 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](#)]
54. Nouiri, M.; Bekrar, A.; Jemai, A.; Trentesaux, D.; Ammari, A.C.; Niar, S. Two stage particle swarm optimization to solve the flexible job shop predictive scheduling problem considering possible machine breakdowns. *Comput. Ind. Eng.* **2017**, *112*, 595–606. [[CrossRef](#)]
55. Vieira, G.E.; Herrmann, J.W.; Lin, E. Rescheduling manufacturing systems: A framework of strategies, policies, and methods. *J. Sched.* **2003**, *6*, 39–62. [[CrossRef](#)]
56. Subramaniam, V.; Raheja, A.S. mAOR: A heuristic-based reactive repair mechanism for job shop schedules. *Int. J. Adv. Manuf. Technol.* **2003**, *22*, 669–680. [[CrossRef](#)]
57. Salido, M.A.; Escamilla, J.; Giret, A.; Barber, F. A genetic algorithm for energy-efficiency in job-shop scheduling. *Int. J. Adv. Manuf. Technol.* **2016**, *85*, 1303–1314. [[CrossRef](#)]
58. Delgado, A.; Jensen, R.M.; Janstrup, K.; Rose, T.H.; Andersen, K.H. A constraint programming model for fast optimal stowage of container vessel bays. *Eur. J. Op. Res.* **2012**, *220*, 251–261. [[CrossRef](#)]
59. Salido, M.A.; Escamilla, J.; Barber, F.; Giret, A.; Tang, D.; Dai, M. Energy-aware parameters in job-shop scheduling problems. In *Proceedings of the GREEN-COPLAS 2013: IJCAI 2013 Workshop on Constraint Reasoning, Planning and Scheduling Problems for a Sustainable Future*, Beijing, China, 3–9 August 2013; pp. 44–53. Available online: <http://ijcai-13.ijcai.org/program/workshop/26> (accessed on 16 November 2019).
60. JSMS CP Optimizer. Available online: http://gps.webs.upv.es/jobshop/data/sched_jobshopEffEnergeticaMkandEnergy.txt (accessed on 16 November 2019).
61. Jiang, T.; Zhang, C.; Zhu, H.; Gu, J.; Deng, G. Energy-efficient scheduling for a job shop using an improved whale optimization algorithm. *Mathematics* **2018**, *6*, 220. [[CrossRef](#)]
62. Data of JSMS Instances. Available online: <http://gps.webs.upv.es/jobshop/> (accessed on 16 November 2019).

