THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Institut National Polytechnique de Toulouse (INP Toulouse)*

Présentée et soutenue le *31/01/2023* **par :**

Igor FONTANA DE NARDIN

On-line scheduling for IT tasks and power source commitment in datacenters only operated with renewable energy

---

JURY

Premier MEMBRE	Professeur d'Université	Rapporteur
Second MEMBRE	Professeur d'Université	Rapporteur
Troisième MEMBRE	Professeur d'Université	Examinateur
Quatrième MEMBRE	Professeur d'Université	Examinateur
Cinquième MEMBRE	Professeur d'Université	Examinateur

École doctorale et spécialité :
MITT : Ecole Doctorale Mathématiques, Informatique et Télécommunications de Toulouse

Unité de Recherche :
Laplace (UMR 5213) et IRIT (UMR 5505)

Directeur(s) de Thèse :
Patricia STOLF et *Stéphane CAUX*

Rapporteurs :
Premier RAPPORTEUR et *Second RAPPORTEUR*

Acknowledgments

Acknowledgments

Abstract

Abstract

Résumé

Résumé

Contents

Abstract iii

Résumé v

1 Introduction 1
- 1.1 Context . . . 1
- 1.2 Problem Statement . . . 2
- 1.3 Main contributions . . . 4
- 1.4 Publications and Communication . . . 5
- 1.5 Dissertation Outline . . . 6

2 Context and Related Work 9
- 2.1 Global Warming and ICT Role . . . 9
- 2.2 Renewable Energy Sources . . . 13
- 2.3 Renewable-only Data center . . . 14
  - 2.3.1 Electrical elements . . . 14
  - 2.3.2 IT elements . . . 17
- 2.4 Sources of Uncertainty . . . 20
  - 2.4.1 Weather Uncertainties . . . 20
  - 2.4.2 Workload Uncertainties . . . 20
  - 2.4.3 Dealing with Uncertainties . . . 21
- 2.5 Literature Review . . . 23
  - 2.5.1 Only Offline Decisions . . . 23
  - 2.5.2 Only Online Decisions . . . 25
  - 2.5.3 Mixed decisions . . . 27
  - 2.5.4 Discussion and Classification of the Literature . . . 28

3 Modelling, Data, and Simulation 33
- 3.1 Model . . . 34
  - 3.1.1 Offline Decision Modules . . . 34
  - 3.1.2 Offline Plan . . . 34
  - 3.1.3 Online Decision Modules . . . 34
- 3.2 Data . . . 34
  - 3.2.1 Workload Trace . . . 34
  - 3.2.2 Weather Trace . . . 34
  - 3.2.3 Platform Configuration . . . 34
- 3.3 Simulation . . . 34
  - 3.3.1 Simulator . . . 34
  - 3.3.2 Metrics . . . 34

3.3.3 Datazero2 Middleware . . . 34
3.4 Conclusion . . . 34

4 Introducing Power Compensations 35
4.1 Introduction . . . 35
4.2 Model . . . 35
4.3 Heuristics . . . 35
4.4 Results Evaluation . . . 35
4.5 Conclusion . . . 35

5 Learning Power Compensations 37
5.1 Introduction . . . 37
5.2 Algorithms . . . 37
5.2.1 Random . . . 37
5.2.2 Q-Learning approach . . . 37
5.2.3 Contextual Multi-Armed Bandit approach . . . 37
5.3 States . . . 37
5.4 Actions . . . 37
5.5 Rewards . . . 37
5.6 Results Evaluation . . . 37
5.7 Conclusion . . . 37

6 Adding Battery Awareness in EASY Backfilling 39
6.1 Introduction . . . 39
6.2 Model . . . 39
6.3 Heuristic . . . 39
6.3.1 Predictions . . . 39
6.3.2 Job Scheduling . . . 39
6.3.3 Power compensation . . . 39
6.4 Conclusion . . . 39

7 Conclusion and Perspectives 41
7.1 Conclusion . . . 41
7.2 Perspectives . . . 41

Bibliography 43

List of Figures

1.1 Problem overview. Online receives an offline plan, the actual renewable production, and the users' jobs. It must define storage usage, job placement in the servers, and server speed. 3

2.1 Estimated global GHG emissions [21]. 10
2.2 Projections of ICT's GHG emissions from 2020 [28]. 11
2.3 ICT's emissions, assuming the 2020 level remains stable until 2050, and global CO2 emissions reduced in line with 1.5°C [28]. 12
2.4 Estimations for global ICT's GHG emissions in 2015 and 2020 [28]. The authors consolidated the works from [11, 13, 59, 60]. 12
2.5 Comparison of small data center load and the generation from a theoretical photovoltaic in Belfort, France. Both load and production have the same average value [81]. 14
2.6 Power consumption on a GRID5000 server when running the same application, but varying the frequency and the number of active cores [39]. 18
2.7 Comparison between FCFS and EASY Backfilling scheduling heuristics. . . 22
2.8 Agent learning process in an environment. At each step, the agent verifies the actual state and chooses an action. The environment executes the action and returns a reward. The agent learns the reward obtained in that state for that action. 23

List of Tables

Chapter 1

Introduction

1.1 Context

Global warming is one of the biggest challenges humanity is facing. A recent rapport shows that we are walking toward a global mean temperature increase by 2100 of 2.7°C, well above the 1.5°C defined by the Paris Agreement [21]. The same rapport predicts the rise in mean global temperature will be around 1.8°C even after implementing all announced Paris Agreement goals. Achieving 1.5°C demands an engagement of all sectors to reduce greenhouse gas (GHG) emissions. GHG is generated during the combustion process of fossil fuel, one of the world's main sources of energy production [67].

One significant GHG emitter is the Information and Communications Technology (ICT) sector. It produces around 1.8-2.8% of the world's total GHG [28]. Inside ICT, Data centers and transmission networks are responsible for nearly 1% of global energy-related GHG emissions [45]. The data center sector is one of the most electricity-expensive ICT actors due to its uninterrupted operation. A report revealed that Google data centers consumed the same amount of energy as the entire city of San Francisco in 2015 [49]. In addition, the situation tends to get even worse due to the improvements reduction in processor technologies and the predicted expansion of internet usage [19, 28].

Big cloud providers such as Google and Amazon are trying to reduce energy consumption and increase the power coming from renewable sources (RES) [61]. RES is the most encouraging method to eliminate fossil fuel use [67]. Renewable sources generate energy from clean sources such as biomass, hydropower, geothermal, solar, wind, and marine energies [5, 12, 33, 70, 81]. A significant drawback of RES is the weather conditions dependency, creating power intermittence. These providers smooth this intermittence by not migrating entirely to RES, maintaining a connection to the grid [81]. Therefore, they are not 100% clean. A renewable-only data center must consider this intermittence in its decision-making. Another source of uncertainty comes from the user's demand. Users can send their requests at any time. Providing high availability is a challenge for a renewable-only data center.

A way to reduce the impact of RES power production intermittence is by adding storage elements [81]. Batteries and hydrogen tanks can shift generation and/or consumption over time. A renewable-only data center demands a massive storage capacity [81]. For example, Google plans to use energy from 350 MW solar panels connected to a storage system with 280 MW [16]. While helping to deal with RES intermittence, storage management introduces another level of decision. For example, it can store energy during the day using at night. Nevertheless, the demand during the day could be higher than at night, so maybe it is better to use the energy during the day. This is another big challenge for migrating

to a 100% clean data center.

Some works propose ways to deal with both demand and weather uncertainties using predictions [31, 35, 57, 96]. Forecasting the upcoming requests and the weather helps to plan storage usage. They use these predictions to maximize renewable usage but with the grid as backup. All these works are valuable and important to optimize renewable usage. However, the forecast can vary from the actual values. Other works focus on reacting to real events [18, 38, 55, 82]. They try to minimize the data center operational cost, maximize renewable usage, increase the revenue of job execution, or improve the Quality of Service. Usually, they define ways to schedule the jobs, optimizing their objective. However, they focus on short-term decisions without long-term management. Since these works also have the grid as backup, storage management is not a concern. Some works mix predictions with reactive actions. For example, Goiri et al. [31] propose a scheduling algorithm that predicts solar power production and uses it to define the best moment to start new jobs, using brown energy (from the grid) when necessary. Also, Venkataswamy et al. [92] created a job scheduler that defines job placement according to the available machines. The available machines are given by a fixed plan (which can use power from renewable, batteries, or grid), with no modifications.

Few research initiatives are investigating how to design and operate a renewable-only data center. One of them is the ANR Datazero2 project [75]. This project aims to define a feasible architecture to maintain a renewable-only data center. This architecture includes several elements to provide energy to the IT servers, such as Wind turbines, Solar panels, Batteries, and Hydrogen tanks. Considering the decision-making, Datazero2 divides the problem into two parts: offline and online. The offline module predicts power demand and production. Using these predictions and considering long-term constraints, this module creates a power and IT plan for the near future.

The online module schedules the users' jobs, using the offline plan as a guide. Online is the only one that knows exactly the jobs submitted to the data center. So, it needs to place them in the available servers. Online could just apply the offline plan without modifications. However, this behavior would impact the Quality of Service (QoS). Online can improve the QoS, increasing storage usage to turn on more servers (to run more jobs) or speed up the running servers (to finish jobs earlier). Also, online must be renewable production aware. For example, online can identify a lower production that can dry the storage faster, so it must reduce its usage. Finding a good trade-off between QoS and storage management is even harder in online mode, which demands fast decisions. In this thesis, we focus on these online decisions. The goal is to design and prove the efficiency of a novel approach for scheduling users' jobs, finding a good trade-off between QoS and storage management.

1.2 Problem Statement

A data center powered by renewable energy demands several levels of decision. Several works aim to optimize some of these decisions. We can cite demand and production predictions, cost optimization, sizing, shifting demand, battery management, admission control, and job scheduling, to mention a few. Usually, these works introduce a link to the grid, using it as a backup to cope with peak demand. Removing the grid of the context adds several challenges. This context increases the need for predictions to manage weather and workload uncertainties. Another key element in renewable-only data centers is storage. Aligning prediction and storage elements allows it to define the best strategy to handle users' requests. However, actual demand and production can vary from the predictions.

So, the online module must react to the actual values. This reaction can improve the QoS (e.g., when there is more production than expected) or reduce the impact of critical events.

Figure 1.1 illustrates all the elements in the decision process. We consider only renewable sources and storage elements without grid connection. An offline optimization gives an offline plan using production and demand prediction. The offline plan has a limited size named time window (e.g., three days). So, offline suggests actions to online during this time window. Online receives the actual renewable production from wind turbines and solar panels. Online adapts storage usage according to the actual production. Since hydrogen has a longer start-up time, it is difficult to manage it in online mode. Therefore, we let hydrogen usage from the offline optimization, using it to provide energy during periods with low renewable production (e.g., during the winter). So, online decides about battery usage only.

Battery management introduces two new challenges regarding the Battery's State of Charge (SoC). SoC means the level of charge of a battery relative to its capacity. A good practice to extend the battery's lifetime is to avoid drying or overcharging it [97]. So, maintaining the SoC between reasonable levels is the first challenge. Online has the entire time window to make modifications in battery usage. However, it must finish the time window close to the expected SoC (given by the offline plan). This is the second challenge. Since the data center runs continuously, it is not viable to always use more battery than expected for every time window.

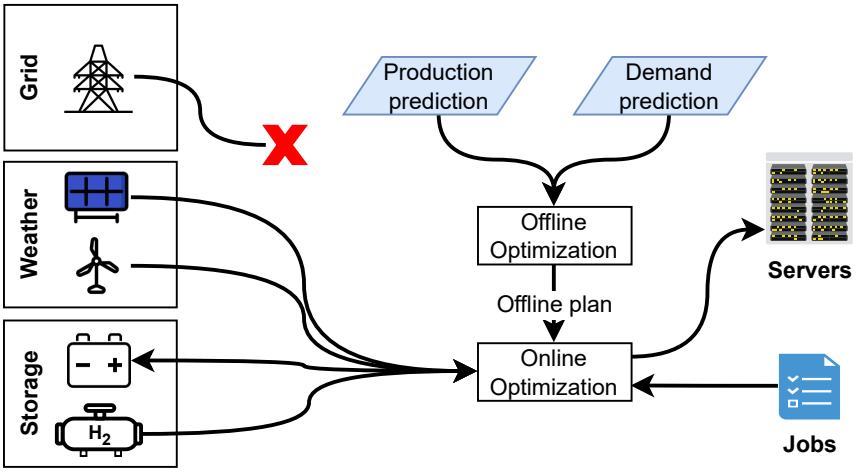

Figure 1.1: Problem overview. Online receives an offline plan, the actual renewable production, and the users' jobs. It must define storage usage, job placement in the servers, and server speed.

On the IT side, online receives the jobs from the users and must schedule them on the available servers. Online receives an offline plan for server configuration (machine on/off and speed). However, it can modify the server configuration to react to incoming events (e.g., more production, demand peak). Changing the speed of a server is possible due to the Dynamic voltage and frequency scaling (DVFS) technique. DVFS allows servers' speed reduction, spending less energy. However, putting a job on a server with a decreased speed can impact the job QoS. To sum up, online must manage the battery (maintaining the SoC between thresholds and finishing the time window with the battery level close to the target), schedule the jobs, and balance the servers' speed.

This thesis' first objective is to make online modifications in the power decisions given

by the offline plan, coping with the uncertainty coming from renewable production and workload demand. The second goal is mixing power and scheduling online decisions, turning the scheduling storage aware. This mix allows the scheduling to make better decisions than usual algorithms. The last objective is to add the predictions to the online decision. These goals help to find a better trade-off between QoS and storage management. Different contributions address these questions in this manuscript.

1.3 Main contributions

Proposing a simulation environment

A crucial step to simulate data center management is defining the workload, weather, server configuration, and simulation tool. We detail in Section 3 the simulation environment, providing a framework for future works. Regarding the workload, some traces are used in literature, such as Google [78], Parallel Workloads Archive [27], and Alibaba [94]. We propose a trace from Parallel Workloads Archive named Metacentrum [50]. We detailed the filtering process of this trace. Considering the weather, it is possible to collect data from everywhere in the world. We present the methodology to generate power production from a NASA trace, using the framework Renewables.ninja[1] [74]. The third input is the server configuration. We demonstrate the data collected from a server in GRID5000[2] used in this thesis. Finally, we present the simulation tool named BATSIM[3], based on SIMGRID[4]. We introduced in this simulation tool the modifications needed to manage battery and power production. The ensemble of these data and definitions allows future work inside and outside the Datazero2 project.

Defining offline power and IT decisions

As illustrated in Figure 1.1, an important part is the offline plan. This plan must consider the power and demand predictions to define the actions for the next time window. We demonstrate in Section 3 a model to use both predictions. We separate the problem into two parts. First, we present the optimization problem to define power engagement, giving a power prediction. This optimization problem results in expected renewable power production, storage usage, and expected SoC. The sum of the expected renewable power production and storage usage is named the power envelope. The second part is the IT servers' state (on/off) and speed definition. This optimization problem defines the state and speed according to the power envelope. The objective of this optimization problem is to maximize the servers' speed. The results of both optimizations are the input for the online module.

Reacting to power fluctuations

Given the result of the optimization problem, next, we propose a heuristic to react to the power fluctuations. Since there is no perfect prediction, one source of divergence is the difference between the prediction and actual values. This divergence occurs in both power demand and production. Also, the offline model considers that the servers will maintain constant power usage. However, the server consumption can vary according to

[1] https://www.renewables.ninja/
[2] https://www.grid5000.fr
[3] https://batsim.org/
[4] https://simgrid.frama.io/

the scheduling and/or job. Yet, the scheduling can modify the battery usage to improve the QoS (e.g., avoiding killing jobs). Considering all these sources of power fluctuations, the heuristic must adapt the usage, aiming to approximate the state of charge of the target level at the end of the time window. Since this is an online problem, we can not re-run the offline optimization solution with the actual values. Therefore, we propose four policies to compensate for these divergences in the power envelope. Each one finds a different moment in the future to place the compensation.

Learning the actions to deal with power fluctuations

The four compensation policies apply the same behavior throughout the entire execution. However, different moments inside the time window can demand distinct policies. So, our next goal is to learn when to use each policy. So, we introduce two Reinforcement Learning (RL) algorithms to discover the best mix of policies. Considering each policy as RL's action, we present the RL's state and reward. The premise of applying RL is that optimizing the decisions locally generates a global optimal. In other words, if the algorithm chooses the best action each time, in the end, we will have the best results. We implemented two well-known RL algorithms named Contextual Multi-Armed Bandit and Q-Learning. We present the learning results and a comparison between the RL algorithms and random choices.

Defining storage-aware scheduling using production and demand predictions

Finally, the last contribution is a storage-aware scheduling heuristic. This algorithm is based on the well-known EASY-Backfilling. The algorithm is named BEASY. BEASY uses the predictions given by the offline to predict dangerous moments, where it must be careful in the scheduling. Also, we introduce another level of validation, verifying if the servers allocated to the job would be available during the entire execution. Regarding power compensations, it creates several possible scenarios of production and demand using the forecasts. According to these scenarios, the heuristic finds the best moment to make the compensations. For example, BEASY tries to reduce the usage before the moments when the predictions indicate that the battery could be lower than a critical value. This heuristic mixes all decisions providing a well-balanced answer to the online multi-objective problem.

1.4 Publications and Communication

Submitted Peer Reviewed Conferences:

- I. F. de Nardin, P. Stolf and S. Caux, "Adding Battery Awareness in EASY Backfilling", 2023 IEEE 35th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD), Porto Alegre, Brazil, 2023.

Accepted Peer Reviewed Conferences:

- I. F. de Nardin, P. Stolf and S. Caux, "Analyzing Power Decisions in Data Center Powered by Renewable Sources", 2022 IEEE 34th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD), Bordeaux, France, 2022, pp. 305-314;
- I. F. de Nardin, P. Stolf and S. Caux, "Evaluation of Heuristics to Manage a Data Center Under Power Constraints", 2022 IEEE 13th International Green and Sustainable Computing Conference (IGSC), Pittsburgh, PA, USA, 2022, pp. 1-8;

- I. F. de Nardin, P. Stolf and S. Caux, "Mixing Offline and Online Electrical Decisions in Data Centers Powered by Renewable Sources", IECON 2022 – 48th Annual Conference of the IEEE Industrial Electronics Society, Brussels, Belgium, 2022, pp. 1-6;

- I. F. de Nardin, P. Stolf and S. Caux, "Smart Heuristics for Power Constraints in Data Centers Powered by Renewable Sources", Conférence francophone d'informatique en Parallélisme, Architecture et Système (COMPAS 2022), Jul 2022, Amiens, France. paper 7.

Others Disseminations:

- Talk: Analyzing Power Decisions in Data Center Powered by Renewable Sources, GreenDays@Lyon, March 2023.

1.5 Dissertation Outline

The remaining dissertation has the following organization:

Chapter 2 - Context and Related Work: This chapter presents the fundamentals to understand this dissertation. Considering the scope of the topic, the context consists of four parts. First, we introduce the context of global and ICT GHG emissions. Then, we describe renewable energy as an alternative to replace brown energy. After, we explain the usage of renewable to power a data center. Then, we define the uncertainties of weather and workload in a renewable-only data center. This last part also clarifies the importance of using predictions but with an online adaptation. After presenting the context, we introduce a list of works that solve part of our problem, highlighting the existing gaps in the state-of-the-art;

Chapter 3 - Modelling, Data, and Simulation: In this chapter, we describe the model to deal with the several elements that compose a renewable-only data center. Datazero2 creates a division between Offline and Online decisions. We present the model to deal with offline decisions using predicted power demand and production. Then, we demonstrate the output of Offline used by the Online. Finally, we define the Online model, which englobes the job scheduling and modifications in the Offline plan. After describing the model, we explain the source of the different data (e.g., workload, weather, servers) applied in the simulations. We present an explanation of the work done in the traces of the literature. Finally, we present the simulation tools used in this work;

Chapter 4 - Introducing Power Compensations: This chapter describes the proposed algorithm to react to power uncertainties. We created four heuristics to find the best place to compensate for battery changes, which aim to reduce the number of killed jobs and the distance between the battery level and the target level. The results presented are related to the publications [25] and [24];

Chapter 5 - Learning Power Compensations: This chapter presents the idea and the results of the introduction of Reinforcement Learning (RL) in the power compensation problem. We propose two RL algorithms (Q-Learning and Contextual Multi-Armed Bandit) to learn the best moment to compensate;

Chapter 6 - Adding Battery Awareness in EASY Backfilling: This chapter explains a heuristic to mix scheduling and power compensation decisions. This heuristic is based on the EASY Backfilling scheduling algorithm but considers the battery's State of Charge to make better decisions;

Chapter 7 - Conclusion and Perspectives: Finally, in this chapter, we summarize the contributions of this work, providing a discussion about future works.

Chapter 2

Context and Related Work

Contents

2.1 Global Warming and ICT Role 9
2.2 Renewable Energy Sources 13
2.3 Renewable-only Data center 14
2.4 Sources of Uncertainty 20
2.5 Literature Review 23

2.1 Global Warming and ICT Role

Global warming is one of the most critical environmental issues of our day [40]. Global warming is the effect of human activities on the climate, mainly the burning of fossil fuels (coal, oil, and gas) and large-scale deforestation [40]. Both activities have grown immensely since the industrial revolution. The burning of fossil fuels process results in greenhouse gas emissions [67]. Today, fossil fuels are one of the world's main sources of energy production, helping to emit more and more GHG [67]. GHG stays in the atmosphere creating a layer as a blanket over the planet's surface. Without this blanket, the Earth can balance the radiation energy from the sun and the thermal radiation from the Earth to space [40]. However, this human-generated blanket imposes a barrier to the thermal radiation from the Earth, letting it into the atmosphere and heating the planet, working as a greenhouse. All this process works as a greenhouse which is the reason for the name greenhouse gas [40].

This situation brings us to United Nations Climate Change Conference (COP21) in Paris, France, on 12 December 2015. At this conference, 196 signed the Paris Agreement aiming to [65]:

1. Reduce global greenhouse gas emissions substantially, limiting the global temperature increase in this century to 2°C while pursuing measures to limit the growth even further to 1.5°C;

2. Review countries' commitments every five years (through the Nationally Determined Contribution, or NDC);

3. Provide financing to developing countries to mitigate climate change, strengthen resilience, and enhance their abilities to adapt to climate impacts.

These are ambitious but necessary objectives. Since then, countries and organizations have proposed several actions and pledges. However, a recent report indicates that the actual world's effort is not enough [21]. Figure 2.1 shows GHG emission and temperature estimations. We could see that there is a small reduction in emissions increase tendency. Nevertheless, this figure estimates that real-world actions based on current policies will lead to an increase of somewhere between 2.6 and 2.9°C by 2100. This estimation is well above the 1.5°C pursued by the Paris Agreement. Considering the targets proposed by the countries through NDC, the temperature will be around 2.4°C. In a scenario based on NDC targets and submitted and binding long-term targets, the prediction is a temperature of 2°C by 2100, the limit proposed by the Paris Agreement. The report forecasts an optimistic scenario analyzing the effect of net zero emissions targets of about 140 countries that are adopted or under discussion. Even in this optimistic scenario, the estimated temperature would be 1.8°C. The situation tends to be even worst with the gold rush for gas [20]. The report indicates that in 2022 we arrived at 1.2°C warming [21].

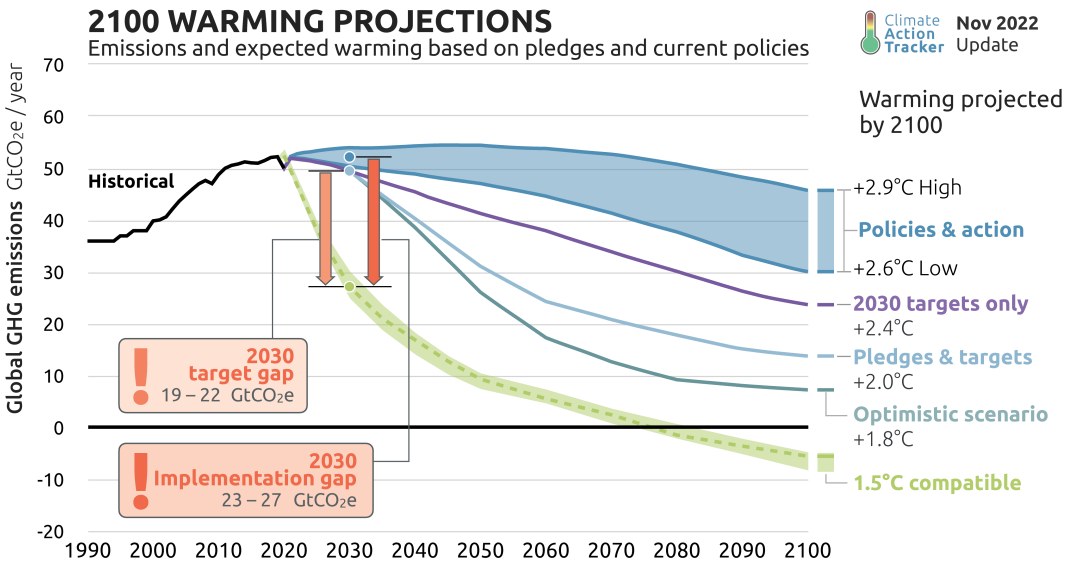

Figure 2.1: Estimated global GHG emissions [21].

We have started to feel the impacts of global warming on humanity, such as heatwaves, droughts, and floods, impacting flora and fauna directly [42, 63]. In a cascade effect, this increases food and water insecurity worldwide [42, 95]. Also, high temperatures increase mortality, impact labor productivity, impair learning, increase adverse pregnancy outcomes possibility, increase conflict, hate speech, migration, and infectious disease spread [52]. Therefore, an increase of the temperature by 2.7°C as forecasted would impact one-third (22–39%) of the world's population by 2100 [52]. Climate change has already impacted around 9% of people (>600 million) [52]. Reducing global warming from 2.7 to 1.5°C results in a ∼5-fold decrease in the population exposed to unprecedented heat (mean annual temperature ≥29°C) [52]. Thus, all sectors must reduce their GHG emissions as much as possible.

Information and Communication Technology is one of these sectors which has accelerated growth in the last 70 years. Unesco defines ICT as [90]:

> "Information and communication technologies (ICT) is defined as a diverse set of technological tools and resources used to transmit, store, create, share or

exchange information. These technological tools and resources include computers, the Internet (websites, blogs, and emails), live broadcasting technologies (radio, television, and webcasting), recorded broadcasting technologies (podcasting, audio and, video players, and storage devices), and telephony (fixed or mobile, satellite, visio/video-conferencing, etc.)."

Regarding the ICT role in GHG emissions, the global share is around 1.8%-2.8%, or 2.1%-3.9% considering the supply chain pathways in 2020 [28]. The situation tends to get even worst, driven by the boom in Internet-connected devices. A Cisco report indicates that the Internet had 3.9 billion users in 2018 [19]. The same report predicts an increase to 5.3 billion in 2023 (66 percent of the global population). Also, they predicted 3.6 networked devices per capita in 2023, up from 2.4 networked devices per capita in 2018. However, International Telecommunication Union (ITU), a United Nations specialized agency for ICTs, indicates that we arrived at 5.3 billion connected users in 2022 due to the COVID-19 pandemic [3]. But will the growth in internet users increase GHG emissions? Andrae and Edler [11] and Belkhir and Elmeligi [13] agree that this growth could lead to an increase in GHG emissions. Figure 2.2 shows the predictions of both works.

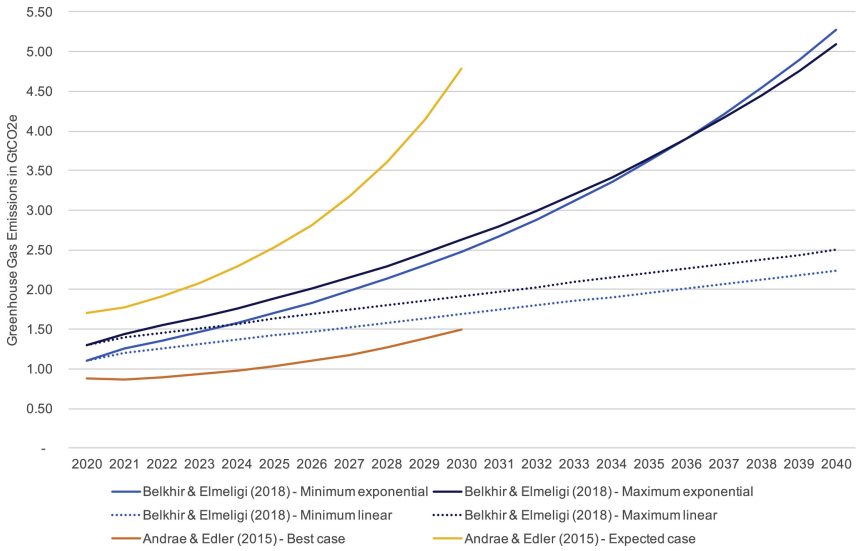

Figure 2.2: Projections of ICT's GHG emissions from 2020 [28].

This figure illustrates the contraction in the Paris Agreement demand and the predictions about usage in the ICT sector. In all forecasts of Figure 2.2, the tendency is emissions growth. However, ICT needs to reduce its emissions drastically. Figure 2.3 illustrates the carbon emission share if the ICT stays at the same level as 2020 and the other sectors decrease their emissions. Without changes, ICT would have 35.1% of global emissions in 2050. So, ICT must move towards reducing its emissions. Figure 2.4 presents the estimations of ICT's GHG emissions for 2015 and 2020 from different authors. This figure breaks down these emissions into different components. One of them, with a good share in some cases, is Data centers. IBM defines the data center as "A data center is a physical room, building or facility that houses IT infrastructure for building, running, and delivering applications and services, and for storing and managing the data associated with those applications and services" [6]. The International Energy Agency (IEA) defines data center as [45]:

"Data centers are facilities used to house networked computer servers that

store, process and distribute large amounts of data. They use energy to power both the IT hardware (e.g., servers, drives, and network devices) and the supporting infrastructure (e.g., cooling equipment)."

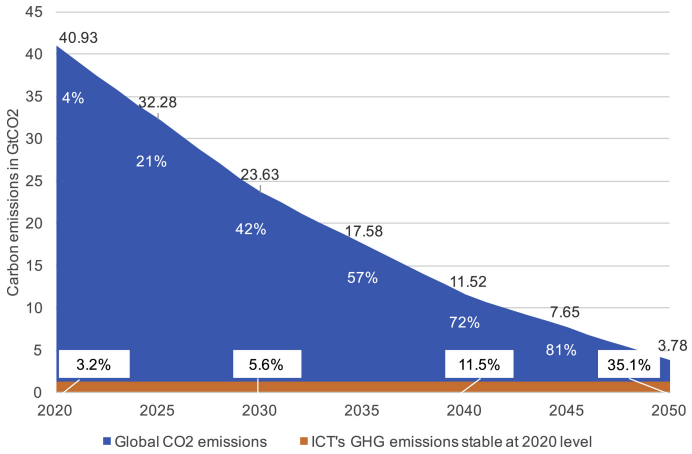

Figure 2.3: ICT's emissions, assuming the 2020 level remains stable until 2050, and global CO2 emissions reduced in line with 1.5°C [28].

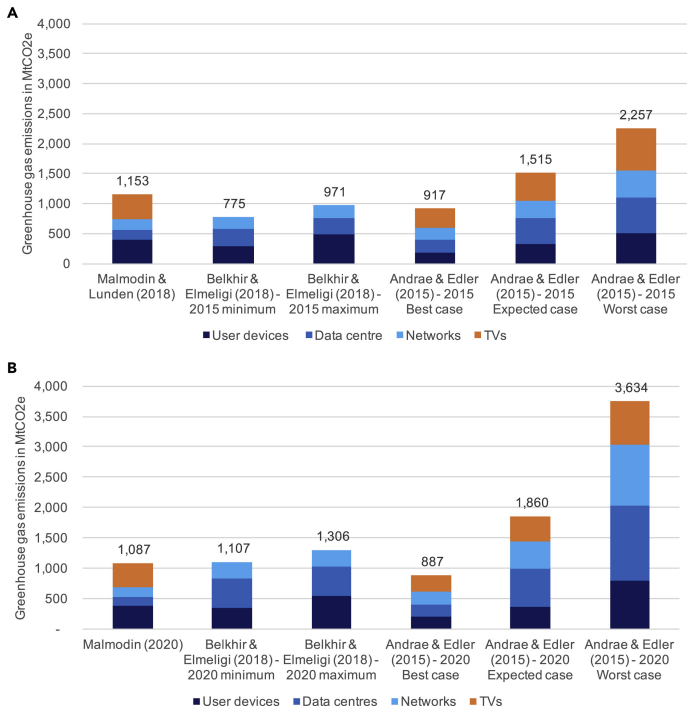

Figure 2.4: Estimations for global ICT's GHG emissions in 2015 and 2020 [28]. The authors consolidated the works from [11, 13, 59, 60].

Data centers are very energy consumers. IEA published an article indicating that data centers and networks were responsible for almost 1% of energy-related GHG emissions in 2020 [45]. Also, Google data centers consumed the same amount of energy as the entire city of San Francisco in 2015 [49]. Global data center electricity use in 2021 was 220-320 TWh, corresponding to 0.9-1.3% of the global demand [45]. For example, the domestic

electricity consumption of Italy was 300 TWh in 2021 [2]. In Ireland, electricity consumed by data centers went from 5% of the total electricity consumption in 2015 to 14% in 2021 [4]. Denmark predicts to triple data center consumption, corresponding to 7% of the country's electricity use [1].

Despite the strong growth in demand, data center energy usage has only moderately grown [45]. A reason that explains it is the improvements in IT hardware energy consumption [45]. These improvements allowed a boost in microchips' speed with a reduction in their power consumption, letting big data center companies cope with the peak in demand. Gordon Moore predicted in 1965 (Moore's law) that [64]:

> "The complexity for minimum component costs has increased at a rate of roughly a factor of two per year. Certainly over the short term this rate can be expected to continue, if not to increase. Over the longer term, the rate of increase is a bit more uncertain, although there is no reason to believe it will not remain nearly constant for at least 10 years."

Even if he predicted it just until 1975, it is the case nowadays. However, the future is uncertain, and the community is divided to confirm continuous efficiency improvements [28]. While Andrae and Edler [11] and Belkhir and Elmeligi [13] expected an ending in power-consuming improvements (indicated in Figure 2.2), Malmodin and Lundén [60] are more optimistic. They suggest that ICT's carbon footprint in 2020 could halve by 2030. To achieve that, he considers two key factors. First, the improvements will continue. Second, the migration to renewable sources.

2.2 Renewable Energy Sources

The ICT migration to renewable energy sources (RES) is one of the factors that helped reduce the growth in GHG emissions despite the rapidly growing demand for digital services [45]. RES is one of the principal solutions to decarbonize electrical production [67, 81]. RES is also named green energy, in contrast to brown energy from fossil fuels. Basically, RES generates energy from natural sources, such as solar, wind, geothermal, hydropower, wave and tidal, and biomass [5, 12, 33, 70, 81]. These natural sources have a low impact on GHG emissions. For example, manufacturing is the stage with higher emissions for wind and solar [10]. So, these components could produce energy with no or low GHG emissions. The renewable term comes from the idea that these sources are constantly replenished. On the other hand, fossil fuels are non-renewable because they need hundreds of millions of years to develop. In the Net Zero Emissions by 2050 Scenario, RES is responsible for one-third of the reductions between 2020 and 2030 [14]. Some countries focus on nuclear power plants to produce energy [51]. Even if nuclear power is a low carbon emissions energy source, it introduces the risk of accidents and environmental impacts of radioactive wastes [51].

The biggest challenge of implementing RES is its intermittence [81]. Since RES production comes from nature, it depends on the climate conditions. For example, there is no power production from solar during the night. There are two approaches to reducing brown usage: on-site and off-site generation [79]. On-site generation uses local renewable resources, and off-site takes resources available on the grid. In an off-site generation, it is not possible to guarantee that the incoming energy is from RES since the grid mixes all types of power generation [81]. Giant cloud providers (e.g., Google, Amazon, and Facebook) invest in solar and wind power plants in an off-site approach [7, 16, 61]. So, they could say that, on average, they provide RES to the grid with the same amount that

they expend. However, they transfer the RES uncertainty problem to third parties [81]. For example, in a case with a peak in demand, they will use the power from the grid, renewable or not. So, they are still non-renewable-dependent.

2.3 Renewable-only Data center

Since data centers have a controlled infrastructure, they are a good target to migrate to a renewable-only environment [81]. However, creating a non-renewable independent data center imposes several challenges. In this kind of data center, all the generation is on-site without backup from the grid. Nevertheless, the production and demand can not match. Figure 2.5 exemplifies the mismatch between the power demanded by a data center and power generation. This mismatch requires a production (electrical) or a load (IT) shift. We will present both electrical and IT elements needed for a renewable-only data center.

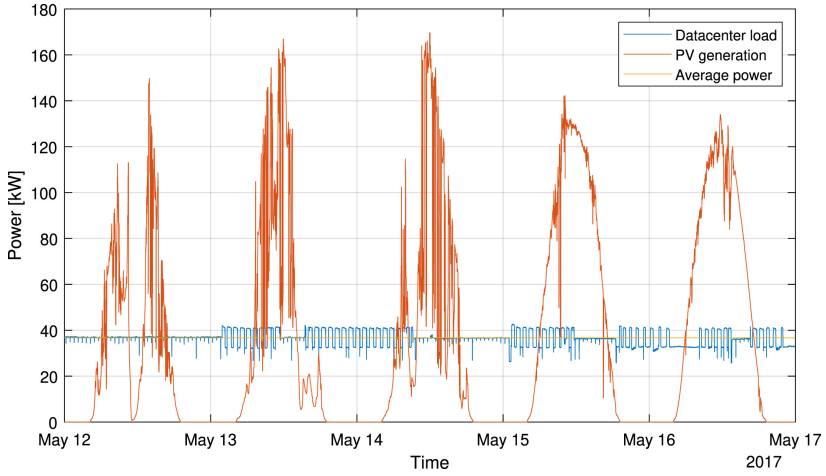

Figure 2.5: Comparison of small data center load and the generation from a theoretical photovoltaic in Belfort, France. Both load and production have the same average value [81].

2.3.1 Electrical elements

As mentioned before, different renewable sources can generate power. We focus on wind and solar since they were the most prominent in the past few years [14]. For wind turbines, the wind speed is crucial. Equation 2.1 gives the power output $P_{WT}(t)$ at the moment t of a wind turbine, given the wind speed v [26, 30, 58].

$$P_{WT}(t) = \begin{cases} 0 & v \leq v_{in} \text{ or } v(t) > v_{out} \\ P_{WT,rated} \times \frac{v(t)-v_{in}}{v_{rated}-v_{in}} & v_{in} < v(t) \leq v_{rated} \\ P_{WT,rated} & v_{rated} < v(t) \leq v_{out} \end{cases} \tag{2.1}$$

Where:

- $P_{WT}(t)$: Power generated by a wind turbine (kW);
- v: Wind speed (m/s);
- v_{in}: Cut-in wind speed (m/s);

- v_{out}: Cut-out wind speed (m/s);
- v_{rated}: Speed related to wind turbine nominal power (m/s);
- $P_{WT,rated}$: Wind turbine nominal power (kW).

If the wind speed v is lesser or equal to the cut-in v_{in} or greater than the cut-out v_{out}, it does not produce power. It tests the cut-out v_{out} to protect the generator. If the speed v is greater than the cut-in v_{in} and lesser or equal to the rated speed v_{rated}, it generates proportionally to the rated power $P_{WT,rated}$ and rated speed v_{rated}. Finally, if the speed v is greater than the rated speed v_{rated} and lesser or equal to the cut-out v_{out}, it produces constant power $P_{WT,rated}$.

Regarding solar production, the photovoltaic (PV) system uses solar panels to generate power from solar irradiance. Equation 2.2 demonstrates how to calculate the output power of a solar panel $P_{pv}(t)$ [26, 58, 84].

$$P_{pv}(t) = P_{R,PV} \times (R/R_{ref}) \times \eta_{PV} \tag{2.2}$$

Where:

- $P_{pv}(t)$: Power generated by each PV panel (W);
- $P_{R,PV}$: PV panel Nominal power (kW);
- R: Solar irradiance (W/m^2);
- R_{ref}: solar irradiance at reference conditions. Usually set as 1000 (W/m^2) [26];
- η_{PV}: PV efficiency.

Regarding PV efficiency η_{PV}, it can consider the temperature of the solar panel [58, 84]. However, some works simplify it by applying a constant value [26, 35]. Equations 2.1 and 2.2 demonstrate that both wind turbines and solar panels depend on wind speed and solar irradiance, respectively. So, the weather conditions drive how much power both can generate.

Due to the weather intermittence, it is necessary to introduce storage elements. These storage elements allow for shifting generation and consumption over time [81]. For example, power coming from wind turbines during the night can be stored and used during the day. Big companies are investing in massive storage elements. An example is Google which is planning a 350 MW solar plant in Nevada connected to a storage system of 280 MW [16]. There are different types of storage with advantages and drawbacks [93]. One of them is hydropower and underground compressed air storage. However, this kind of storage is very geographical, geological, and terrain dependent, which makes it inappropriate to use in data centers [81]. Another type is the very short-term storage such as flywheels or supercapacitors. These storages can output and absorb energy over ms to minutes [93]. They are very suitable for maintaining power stability but not for storing energy for a larger time horizon (e.g., hours or days) [81]. In this thesis, we focus on the batteries and Hydrogen Storage System (HSS).

Batteries are electrochemical devices that store energy in chemical form [81, 93, 98]. They are very reactive because they do not need a warm-up to store/generate power. Batteries are good for short-term storage scenarios (e.g., several hours, day/night cycles) [81]. However, they are inappropriate for longer periods due to their self-discharge rate and low energy density [81, 98]. Historically, Uninterruptible Power Supply (UPS) added batteries

to avoid the server's blackout, doing a soft shutdown that avoids several problems, such as data loss, data corruption, work loss, etc. A problem with batteries is the degradation in capacity and performance over time, requiring battery replacement [81]. A way to extend battery life is by avoiding charging/discharging too extensively [97]. There are some methods to model the energy level inside the battery, such as energy-based, Current-based, or State of Charge [81]. We focus on the State of Charge since it represents the percentage of energy inside the battery according to its capacity (e.g., 100% means battery full and 0% dry). Xu et al. present results showing that maintaining SoC at a narrow range reduces battery degradation [97]. However, using a narrow range would reduce the battery's effectiveness because it can deliver less energy to deal with intermittence. So, the battery SoC must be maintained within a range considering this trade-off. Equations 2.3 and 2.4 demonstrate how to calculate the State of Charge [35].

$$E_{bat}(t) = (E_{bat}(t-1) \times (1-\sigma)) + (P_{ch}(t-1) \times \eta_{ch} \times \Delta t) - (\frac{P_{dch}(t-1)}{\eta_{dch}} \times \Delta t) \quad (2.3)$$

$$SoC(t) = \frac{E_{bat}(t)}{B_{size}} \times 100 \quad (2.4)$$

Where:

- Δt: Duration of t (h);
- $E_{bat}(t)$: Energy in the battery at instant t (kWh);
- $P_{ch}(t-1)$: Charging power (kW);
- $P_{dch}(t-1)$: Discharging power (kW);
- σ: Battery self-discharge rate (%);
- η_{ch}: Battery charge efficiency (%);
- η_{dch}: Battery discharge efficiency (%);
- B_{size}: Battery size (kWh);
- $SoC(t)$: State of Charge at instant t (%);

We can divide Equation 2.3 into three parts. The first part $(E_{bat}(t-1) \times (1-\sigma))$ calculates the natural self-discharge, ignoring charging or discharging the battery. The second part $(P_{ch}(t-1) \times \eta_{ch} \times \Delta t)$ computes the energy stored in the battery according to the charging power. The last part $(\frac{P_{dch}(t-1)}{\eta_{dch}} \times \Delta t)$ is similar but for discharging. Both charging and discharging are not perfect with some losses given by η_{ch} and η_{dch}. For example, if we charge 1 kW this does not mean that, after one hour, we charge 1 kWh. We will charge $1kW \times \eta_{ch}$ (where $\eta_{ch} < 1$). Also, we can not charge and discharge the battery simultaneously, so if $P_{ch} > 0$ then $P_{dch} = 0$, and vice-versa [35]. Equation 2.4 normalizes the SoC to percentage.

Hydrogen, differently from batteries, is more suitable for long-term storage (e.g., over seasons), mainly because it can store large amounts of energy with very low self-discharge [76]. A big limitation of this kind of storage is the lack of reactivity since it demands a longer warming-up time. Also, it includes performance degradation concerns, low efficiency compared to batteries, high costs, and complicated safety measures [81]. Even with all these drawbacks, it is a good solution for storing energy during abundant periods (e.g.,

summer) and using it during lacking periods (e.g., winter). Three elements compose an HSS: electrolyzer, hydrogen tank, and fuel cell. The electrolyzer produces hydrogen from electricity, according to Equation 2.5 [35].

$$P_{ez}(t) \times \Delta t = \frac{HH_{h_2} \times Q_{ez}(t)}{\eta_{ez}} \tag{2.5}$$

Where:

- $P_{ez}(t)$: Power put into electrolyzer (kW);
- HH_{h_2}: H2 higher heating value (kWh/kg);
- $Q_{ez}(t)$: Electrolyzer H2 mass flow (kg);
- η_{ez}: Electrolyzer efficiency (%);

This equation indicates how much hydrogen is added to the tank ($Q_{ez}(t)$) according to the electrolyzer operating power ($P_{ez}(t)$). On the other hand, the fuel cell transforms hydrogen into electricity, according to Equation 2.6 [35].

$$P_{fc}(t) \times \Delta t = LH_{h_2} \times Q_{fc}(t) \times \eta_{fc} \tag{2.6}$$

Where:

- $P_{fc}(t)$: Power delivered by fuel cell (kW);
- LH_{h_2}: H2 lower heating value (kWh/kg);
- $Q_{fc}(t)$: Fuel cell H2 mass flow (kg);
- η_{fc}: Fuel Cell efficiency (%);

Similarly, this equation indicates how much hydrogen is removed from the tank ($Q_{fc}(t)$) according to the output power of the fuel cell ($P_{fc}(t)$). To calculate the Level of Hydrogen ($LoH(t)$ (kg)) Equation 2.7 consolidates the result of the electrolyzer and the fuel cell.

$$LoH(t) = LoH(t-1) + Q_{ez}(t-1) - Q_{fc}(t-1) \tag{2.7}$$

2.3.2 IT elements

While electrical elements are power producers (wind turbines and solar panels) or producers/consumers (storage), the IT elements are entirely power consumers. IT power consumption can be divided into two parts: IT hardware (e.g., servers, data storage, and network devices) and supporting infrastructure (e.g., cooling equipment) [23, 45]. This thesis focus on computing nodes (servers) and scheduling policies on the IT side, so we do not consider data storage, network devices, and supporting infrastructure. There are several articles dealing specifically with these components [23, 37, 69, 100]. The servers are powerful, high-performance machines designed to handle intensive computational tasks and ensure the efficient functioning of various applications and services. They are optimized for reliability, scalability, and performance. Even with these optimizations, they do not have a negligible power consumption [43, 69].

The server power consumption is divided into two parts: static and dynamic [39, 69]. Static power consumption is constant and given by current leakage present in any powered system. Dynamic power is not constant and depends on computing usage. There are

different models to estimate power consumption, such as mathematical linear and non-linear, linear regression, lasso regression, support vector machines, etc. Equation 2.8 expresses a mathematical linear representation of static and dynamic power [39, 43].

$$Pcpu(t) = P^{static} + (P^{dynamic} \times u_{cpu}) \tag{2.8}$$

Where:

- $Pcpu(t)$: Power consumption at moment t (W);
- P^{static}: Static power consumption (W);
- $P^{dynamic}$: Dynamic power consumption (W);
- u_{cpu}: CPU usage (%);

While Ismail and Materwala indicate that P^{static} can be considered as the power idle [43], Heinrich et al. demonstrate a slight difference between the power usage at fully idle and when the real P^{static} [39]. The work of Heinrich et al. is the base for a well-known data center simulator named Simgrid[1] and its evolutions. This article also indicates that $P^{dynamic}$ depends on the application and the server frequency. Figure 2.6 shows the linearization of the power consumption according to the frequency for the same application. Setting different frequencies is possible through the Dynamic Voltage and Frequency Scaling technique. Putting the server at a lower frequency reduces the server's power consumption (as illustrated in Figure 2.6). However, it also decreases the server's speed. Nevertheless, DVFS is a possible solution to reducing energy consumption in moments with lower power available.

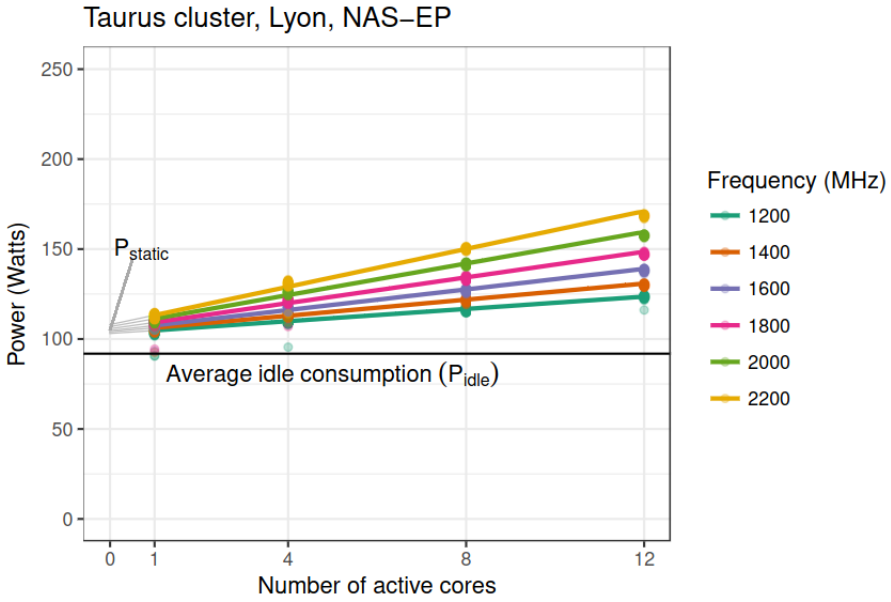

Figure 2.6: Power consumption on a GRID5000 server when running the same application, but varying the frequency and the number of active cores [39].

Another possibility, more drastic, is putting the server to sleep. In the sleep state, the server is unavailable but consuming its lower power possible. Besides being inaccessible,

[1] https://simgrid.org/

another consideration in the sleep state is that sleep transitions (on→off and off→on) are not instantaneous and waste energy. Raïs et al. present a Dynamic Power Management (DPM) solution [77]. This DPM estimates a T_{wait} threshold that when the server is idle for more than T_{wait} seconds, it is more energy-efficient to switch the server off. Equation 2.9 represents their idea.

$$T_{wait} = \max(\frac{E_{OnOff} + E_{OffOn} - (P_{Off} \times (T_{OnOff} + T_{OffOn}))}{P_{idle} - P_{Off}}, (T_{OnOff} + T_{OffOn})) \tag{2.9}$$

Where:

- T_{wait}: Waiting time before putting the server to sleep (s);
- P_{idle}: Power consumption when the server is unused, but powered on (W);
- P_{Off}: Power consumption when the server is off (not null and lower than P_{idle}) (W);
- E_{OnOff}: Energy consumed during the On→Off transition (J);
- E_{OffOn}: Energy consumed during the Off→On transition (J);
- T_{OnOff}: Time spent by the server on On→Off (s);
- T_{OffOn}: Time spent by the server on Off→On (s);

A data center's main objective is to execute users' applications. Running applications in the servers variates the server's CPU usage u_{cpu}. Data centers receive plenty of different application types. We can separate these applications into two big categories: services and batch [22, 81]. Services are applications that interact with different clients. These clients make requests answered by a running service. Each request has a low processing time, but the ensemble of these requests can be very CPU-consuming [62]. Also, the service must answer the request as soon as it arrives. On the other hand, batch applications (or parallel jobs [27]) do not run interactively. While services can run indefinitely, batches have a start and end time. Usually, these applications aim to solve complex problems, such as weather prediction, optimization problems, and simulations, being very long and CPU-consuming [62]. Batch jobs are more flexible considering the moment to execute them, allowing the batch scheduler to define the best moment in the future to run them. Both services and batches demand different approaches and algorithms to deal with them. This thesis focuses on batch high-performance computing (HPC) applications. An HPC job is composed by [81, 86, 88]:

- Submission time: The moment when the user sends the job;
- Requested resources: The resources demanded by the job, such as the number of cores, servers, memory, etc;
- Estimated execution time or walltime: The user indicates how long the job executes. If the real execution time is equal to the walltime, the scheduler kills the job.

2.4 Sources of Uncertainty

After describing renewable-only data center elements, in this section, we detail the sources of uncertainty. First, we start presenting the uncertainty from electrical components due to weather conditions. After that, we describe the uncertainties from server power consumption and HPC jobs. Finally, we discuss the challenges in dealing with all these uncertainties.

2.4.1 Weather Uncertainties

As presented in Section 2.3.1, the objective of the electrical components (solar panels and wind turbines) is to generate power. So, they transform natural renewable resources into energy. Due to the intermittence of these renewable resources, the output power is also intermittent [73]. Regarding solar panels, the output power is calculated easily, using Equation 2.2, in a "clear-sky" condition [89]. "Clear-sky" considers an exposition total of the panels to the sun. However, solar irradiance is impacted by several weather conditions, such as clouds, aerosols, and other atmospheric constituents [89]. Also, the panel efficiency is temperature dependent. Concerning wind turbines, the power output depends on the wind speed (see Equation 2.1). The production has lower and higher wind speed thresholds, meaning that even too slow/fast wind will not produce power.

Due to the renewable intermittence, it is crucial to forecast weather conditions to estimate future power production. Several works propose ways to predict these conditions [83, 85, 87, 89]. Two key terms are important in renewable production: Predictability and Variability [73, 87]. Predictability means the ability to anticipate the availability of a generation resource [73]. For example, solar irradiance is more predictable than wind speed because the forecast accuracy on clear days is high, and satellite data tracks precisely the direction and speed of clouds [73]. On the other hand, due to the erratic nature of the atmosphere, there is randomness in wind power production [83]. Variability indicates the variation over time in production [73]. Both wind and solar can vary. For example, the wind has high variability because it will deviate from 0%–100% over a day [73]. Another element that influences forecast accuracy is the time horizon. For example, the next five minutes are more predictable than the next three days.

2.4.2 Workload Uncertainties

Workload uncertainties come from two sources: the server's energy consumption and jobs. Estimating the real power consumption of a server is not trivial. Several works try to find a model to describe energy consumption or even apply machine learning to define it [23]. Even two machines with the same configuration can consume differently [68]. It is also true that each application can have a completely different energy consumption, mainly because they use the CPU differently [68]. Equation 2.8 presents a simplification of server power consumption. However, this equation is still applicable since different servers can have different dynamic ($P^{dynamic}$) and static (P^{static}) power. Also, considering that energy consumption is Equation 2.8 integral, different applications can have distinct CPU usage (u_{cpu}). Even if the equation is still appropriate, defining its parameters is challenging. For example, the CPU usage (u_{cpu}) of a job can vary between executions (e.g., due to different application parameters). Also, new applications do not have records to estimate their usage. Considering the static power (P^{static}), it is known that it can vary according to the processor's heat [71].

Besides impacting server consumption, jobs have their own uncertainties. A workload

(ensemble of jobs) can be predicted as a load mass or resource usage (e.g., CPU usage over time) [62, 91]. These predictions indicate the estimated demand load, but the exact jobs' arrival is very difficult to predict. The submission is one of the job uncertainties. In a renewable-only data center, this uncertainty mainly impacts the number of servers available. For example, if a server is available expecting a job, but the job does not come or arrives late, this server wastes energy unnecessarily (e.g., by being idle, turning on/off). The second job uncertainty is the execution time. The scheduler receives jobs with requested resources and walltime. So, the scheduler will find a placement for each job to match the requested resources during the walltime. The walltime is a user expectation of the execution time that can be overestimated [88]. An overestimated walltime reduces the effectiveness of the scheduler because it will reserver more time than necessary for the job [86, 88].

2.4.3 Dealing with Uncertainties

After describing the uncertainties in electrical and IT elements, we present some ways to deal with them. The renewable-only data center global problem is a scheduling problem under power constraints. Therefore, the problem includes:

- finding the best moment to start jobs;
- increasing power production from energy storage to improve QoS (e.g., running more jobs, avoiding killing jobs, finishing jobs earlier, etc.);
- adapting power consumption to dealing with over/underproduction;
- starting/stopping servers matching the defined power consumption;
- letting battery between the safe state of charge thresholds.

An optimization problem must consider all these elements. We can divide the problem into offline and online. Offline optimization uses predictions (from weather and workload) to optimize the decisions. Some methods are available to estimate power production and demand, such as Artificial Neural Networks, Support Vector Machines, Markov Chains, Regression Models, Autoregressive Models, and a combination of the methods, such as using genetic algorithms to optimize a neural network [62, 83, 87, 89, 91]. Then, this optimization finds the best approach to match production and demand (e.g., shifting the load, using more power from batteries, rejecting jobs, etc.). Finally, the offline optimization result is applied to the real scenario of production and demand. The idea is to show that even under the uncertainties, the optimized result is good enough. However, offline optimization does not react to real events. For example, it maintains the plan even in a scenario with under/overproduction. Also, the power demand for the workload is treated as a mass, even if in practice a data center receives jobs. This workload simplification helps to solve the optimization problem since the scheduling problem is an NP-Complete [8, 80]. Some works propose offline scheduling, knowing all information from the jobs. However, this is unrealistic in reality [80].

On the other hand, online optimization does not know any future events (e.g., job arrival and power production), discovering them on the fly. Since online just knows actual events, it can not find the optimal global solution. So, online reacts to the incoming events optimizing the problem locally. The online must solve the problem fast because the system can not wait too long for an answer. To sum up both online and offline: offline uses predictions to optimize, but it is not reactive; online is future-blind, just reacting to

actual events. Then, a third possibility emerges: A mix between offline and online. This combination allows taking the best from each side (prediction and global optimization from offline and reactivity from online).

There are several methods to optimize this problem. We can divide them into four groups: (i) exact algorithms; (ii) greedy heuristics; (iii) machine learning; and (iv) meta-heuristics.

The exact methods consist of creating a mathematical model of the problem. The model defines an objective function. It is possible to optimize the objective function through Linear Programming (LP). Solvers such as CPLEX[2] and Guroby[3] are used to find the optimal. The drawback of this approach is its high computation time in large problems, especially if one or more variables are integers (called Mixed Integer Linear Programming - MILP). So, it is not suitable for online optimization, but it is the best approach for offline (when the solving time is not a constraint).

A greedy heuristic is a problem-solving strategy employed in algorithm design that aims to efficiently find approximate solutions by making locally optimal choices at each step, without considering the overall global optimality of the solution. This heuristic operates by iteratively selecting the most advantageous option based on defined criteria or objective functions. Although it may not guarantee the optimal solution, the greedy heuristic's simplicity and computational efficiency make it particularly useful for tackling large-scale problems. Two examples of heuristics for job scheduling are First Come First Served (FCFS) and Easy Backfilling. Figure 2.7 demonstrates the differences between both algorithms. In FCFS, the jobs are placed in the order they arrive. The Easy Backfilling approach tries to fill the hole in scheduling with small jobs (J4 in the figure). Easy Backfilling is highly dependent on walltime estimation in this backfilling step [86, 88].

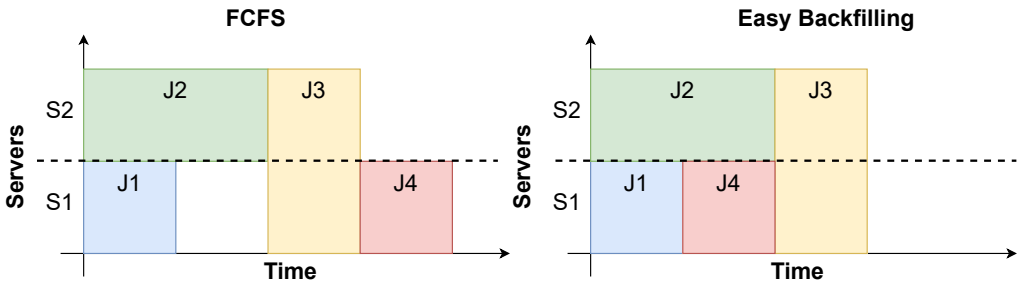

Figure 2.7: Comparison between FCFS and EASY Backfilling scheduling heuristics.

Machine learning is a subfield of artificial intelligence that contains algorithms capable of automatically learning from data and improving performance on specific tasks. In some cases, they emulated the process of human learning. For example, Artificial neural networks simulate the neural network from the human brain. Another example is Reinforcement Learning (RL) which considers the trial-and-error approach, where an agent explores an environment, takes actions, and receives feedback [44]. Figure 2.8 illustrates this process. Through this iterative process, the agent learns to adapt its behavior by optimizing a policy. The algorithm reinforces (from where the Reinforcement Learning name comes) good actions. At some point, the algorithm stops exploring the environment and starts to use its knowledge from previous explorations to repeat what gave the best rewards.

[2]https://www.ibm.com/fr-fr/analytics/cplex-optimizer
[3]https://www.gurobi.com/

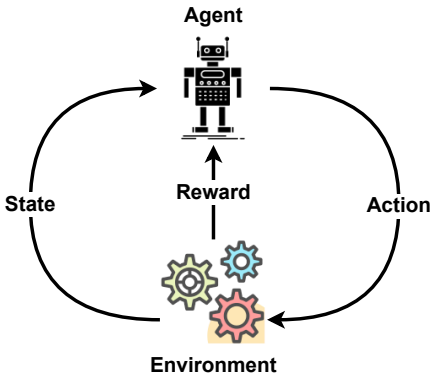

Figure 2.8: Agent learning process in an environment. At each step, the agent verifies the actual state and chooses an action. The environment executes the action and returns a reward. The agent learns the reward obtained in that state for that action.

Finally, metaheuristics are another kind of algorithm to solve hard optimization problems. The "meta" term indicates that they are "higher level" heuristics, different from the problem-specific heuristics [15]. They are nature-inspired (based on some principles from physics, biology, or ethology). An example is the Genetic algorithm which simulates the evolution and mutation process from biology. Another example is Swarm algorithms inspired by the collective behavior of social insect colonies and other animal societies. Generally, metaheuristics are used to solve problems with no satisfactory problem-specific algorithm [15].

In this thesis, we apply all previous methods but metaheuristics. In the offline part, we applied exact algorithms to find optimal solutions. For online, we propose heuristics to solve the specific problem. We also attempted to introduce RL to learn the environment's behavior. Since the online problem needs a fast solution for a specific problem, metaheuristics were not studied.

2.5 Literature Review

This section presents works to solve the issues related to a renewable-only data center. We verify the decisions on both offline and online levels. Some articles are not renewable only but introduce renewable in the decision process. We divided the articles into three groups: only offline, only online, and mixed decisions. The works are presented in chronological order inside each group.

2.5.1 Only Offline Decisions

Gu et al. [34] proposed an Integer Linear Programming for minimizing the carbon emissions of a data center while meeting the scheduling QoS (response time). They work with distributed data centers. Their LP finds the optimal solution to meet a response time constraint, the minimal number of servers, and the best moments to make the server available (e.g., when there is more renewable production). They used an M/M/n model to schedule service requests. Also, they inserted an electricity budget as a constraint. Kassab et al., in [46], [47], and [48], defined an offline scheduling model to minimize the makespan under power constraints (renewable-only data center). The works were developed in the context of the Datazero project. They proposed heuristics and metaheuristics to find the

optimal solution to the NP-Hard scheduling problem. The works [46] and [47] focus only on schedule, maintaining the server availability static (e.g., the server state is previously), while in [48], the authors take a step further, adding server decisions (machines on/off). The server decisions are to turn on servers when needed (and there is power available in the power envelope) and shut down idle servers. All three works ignore power decisions, such as using more or less power from batteries, or online uncertainty.

In [41], the authors created a weighted average scheduling algorithm (WALECC). This algorithm receives a DAG of a parallel application and defines the job placement. This article is unrelated to renewable energy. However, the problem of a system with energy constraints is similar to a renewable-only data center. WALECC mixes heterogeneous processors with DVFS decisions to find the optimal placement (considering makespan) respecting the energy constraints. WALECC works offline because it knows all the jobs in advance, having the information of each job execution time on each processor/speed. Since it receives the relationship between the jobs through a DAG, it respects the jobs' precedence. Lu et al. [56] presented a robust optimization for scheduling and power decisions aiming to minimize energy costs. Their scheduling model knows the execution time of each job at different nodes. The authors introduced renewable production uncertainty in the model. Since the uncertainty introduces a random variable, they created a threshold-based algorithm to choose the solution considering the uncertainty distribution. The cost minimization is solved using Linear programming on MATLAB's MOSEK optimization toolbox.

Caux et al. [17] introduced RECO, a Genetic based algorithm. RECO aims to minimize jobs' due date violations in a renewable-only data center. In an offline way, RECO defines the optimal DVFS frequency to run batch jobs. Also, it decides server states (on/off). Therefore, it works on both scheduling and server configuration. RECO applies a genetic algorithm, creating several scheduling possibilities (pair of job and server) as chromosomes. It applies crossover and mutation, selecting the best-fitted genes. They proposed two fitness functions (to choose the genes). First, a function to reduce the number of due date violations, and the second one uses a weight-based approach, mixing due date and power consumption. The exact processor, frequency, and starting time are assigned using a greedy heuristic. [18] is an evolution of [17], where Caux et al. proposed a new heuristic named MinCCMaxE and a heuristic for degradation. The objective is to maximize the profit from the batches and services execution. Both services and batches are composed of different phases. Services run all the time window duration. On the other hand, MinCCMaxE must find the best placement for batches. MinCCMaxE cross-correlates task and processor loads (both as time series). If it does not find placement (due to power constraints), it delays or degrades the jobs. The degradation step considers the impact on the profit. Both works ([17] and [18]) were developed in the context of the Datazero project.

Haddad et al. [35] modeled a Constraint Satisfaction Problem to define power decisions in a renewable-only data center. This work considers wind and solar as renewable production and battery and hydrogen as energy storage. The objective is to find the power decisions (e.g., battery charge, hydrogen discharge, etc.) that approximate the power produced and demanded. They defined target levels of battery and hydrogen hard constraints. So, the decision variables are battery and hydrogen usage and a relax factor applied to the relationship between produced and demanded. The model considers all losses in power generation, such as battery discharge rate, battery charge/discharge efficiency, hydrogen charge/discharge efficiency, etc. This work is also in the context of the Datazero project. In [29], the authors proposed a system for matching renewable production and demand.

They predicted renewable generation and energy demand using Long Short-Term Memory (LSTM). Renewable production comes from wind turbines and solar panels, using brown energy as a backup. The model uses the result of the predictions to map renewable sources to physical machines. They solved the problem using integer linear programming and Deep Q-Network. Deep Q-Network is an extension of the Reinforcement Learning algorithm Q-Learning.

Wiesner et al. created Cucumber, an admission control policy. The authors use probabilistic forecasts to predict power demand (from workload) and production (from renewable sources). Using both predictions, they calculate how much energy is free to use (named freep). The freep is a time series of renewable production minus power demand. Since probabilistic forecasts results in several predictions, they introduced a parameter to tune the forecasts' optimism. Then, they evaluate the freep time series to accept or reject new jobs in an FCFS fashion. They verify if placing a new job using freep capacity would violate the deadline from the other jobs. If so, they reject the new job. The idea is to maximize the peak of renewable production without adding batteries. Yuan et al. proposed an optimization problem to minimize the operating cost of the entire data center. The data center is powered by renewable energy coming from wind turbines and solar panels, energy storage, and the grid. Their model considers the possibility of selling and buying energy to the grid. For solar and wind turbines, the authors take into account the operation and maintenance costs. They modeled the IT part considering network equipment and cooling system. Similar to other works, they consider batches and services, considering their delay proprieties.

2.5.2 Only Online Decisions

Aksanli et al. [9] presented an online heuristic that uses short-term (30 min) forecasts for green energy in the scheduling process. The green energy comes from wind and solar. The scheduler has two queues, one for services and another for batch. Services will run independently of the green energy available, using brown if needed. Then, they predict the green energy available in the next period. Using this prediction, they estimate the number of slots available to run batch jobs. If the number is greater than the currently available slots, they schedule new jobs and spread the remainder to the running ones. If this number is smaller, the scheduler deallocates some jobs (reducing slots, killing, suspending them, or using brown energy). The authors compared their predictive heuristic with a reactive scheduler that allocates servers according to the energy available.

In [31], the authors proposed a parallel job scheduler, named GreenSlot, for data centers powered by solar energy but using the grid as backup. Their scheduler uses predictions of solar generation to place jobs in the moments with higher production. If it is impossible to place all the jobs in these moments, GreenSlot finds the moments where the grid energy price is lower. GreenSlot saves energy by deactivating idle servers. It creates several slots with the cost. GreenSlot calculates the cost assuming zero for green and the grid price for brown. It assigns penalty costs on slots that cause the job's deadline, avoiding them. The authors indicated some limitations in their work, such as high job rejection or missed deadlines in data centers with high utilization.

Li et al. [53] created a framework named GreenWorks to manage power and server decisions in a data center powered by renewable energy and using battery and grid as backup. They defined a heuristic to manage power generation in four stages. Stage I is when renewable generation is enough to ensure full-speed server operation. In this stage, the renewable production excess is stored in the batteries. Stage II is active when the production is inadequate to provide the power demanded. Here, the heuristic balance

between discharging the battery and impacting the jobs (through DVFS). If this stage is not enough to handle the power mismatch, the system enters stage III. In this stage, GreenWorks tries to decrease load power more, use UPS energy if it has power, and, the last resource, it use the grid. They only use power from the grid for the same amount of energy that they exported previously. So, they accumulate a budget of net green energy exported. If it is not enough, stage IV shuts down the servers. GreenWorks does not use predictions and relies on the first-come-first-serve (FCFS) scheduler.

The authors in [54] created an opportunistic scheduling heuristic. This heuristic tries to minimize brown energy usage by mixing batteries and solar production. The heuristic takes into account services and batches. When the energy consumption is higher than the solar supply, they suspend batch jobs and consolidate the VMS, switching off the servers. The scheduler takes energy from the battery before going to the grid. Just after the batteries dry, it starts to consume brown energy. They implemented a First Fit Decreasing (FFD) scheduling algorithm. Grange et al. [32] proposed an algorithm named Attractiveness-Based Blind Scheduling Heuristic (ABBSH). ABBSH introduce a negotiation model for electrical and IT systems, where both know only their own model. Negotiation helps to deal with different objectives. For example, the objective of the electrical system is to reduce brown energy usage, while IT is to respect the System Level Agreement (SLA) criteria. Both calculate a normalized metric named attractiveness. This metric represents the quality of a given proposal. So, for each job, it calculates the attractiveness of its placement in the IT and electrical context. Then, a function defines, among all possible placements, which one has the best attractiveness. The authors select the best attractiveness using a simple weighted sum of both (electrical and IT), a weighted sum of the hyperbolic sinus of both, or a fuzzy-based one. In this work, the authors consider the electrical part as a black box without making power decisions.

Haghshenas et al. [36] developed a heuristic aiming to minimize energy costs. The energy cost (from the grid) considers IT and cooling usage not provided by solar generation. This heuristic considers services and batches. It always schedules the services in the FIFO approach. It finds the best moment to place batches using best-fit and considering solar production and energy price. Even online, the authors assume knowing all the jobs for the next period. They used solar production predictions to make decisions for the next time slot. They consider the possibility of selling solar production to the grid. Also, they do not consider power on/off transitions, assuming the IT energy consumption is zero when the servers are not running jobs. Finally, the authors updated their algorithm, adding a simplified battery model. The battery will store the surplus solar generation to use later, reducing the energy cost. [66] proposed an online heuristic to schedule jobs in servers. The main objectives are to minimize the makespan, energy cost, and overall cost and maximize renewable usage. The authors separated the heuristic into three phases. First, they estimated the completion time and cost for the execution of a user request on each data center. The cost considers if the data center is powered by renewable or non-renewable. The second phase calculates the fitness value, normalizing completion time and cost. Finally, the last phase takes the data center with higher fitness to place the user request. The work considers that all the data centers are available all the time (using renewable or not).

In [38], He et al. created an online scheduling heuristic to minimize the energy cost of a data center called ODGWS (Online workload Scheduling algorithm with Delay Guarantee). The authors take into account solar, wind, and grid energy. They simplified the job description to be the power requested at each time step by services and batches. The scheduler must deliver the services' power requested in the same step that they arrive.

On the other hand, the scheduler can delay the batches' power demand until a fixed time horizon. So, even if the authors named their algorithm workload scheduling, the decisions are which source to use to provide the energy to the jobs. They do not consider the placement problem (e.g., which server will receive the jobs). Also, they assume that the servers are available all the time. Their problem is a constrained stochastic problem solved by dynamic programming, but they translated it into an online heuristic, which can work without knowing future events.

Peng et al. designed REDUX3 an energy management system with a renewable-aware scheduler. REDUX3 uses energy from different sources, such as wind turbines, solar panels, the grid, diesel generators, and batteries. The system focuses on batch tasks, allowing them to postpone jobs to match production and demand. They added the grid to deal with uncertainties. They created three energy states: Outage Case, Stable Case, and Fluctuate Case. An Outage Case is when the renewable supply is at the minimum, a Stable Case is when the renewable supply exceeds the maximum level, and a Fluctuate Case is when the previous energy state was Outage Case but is stable now. Also, they introduced three key components. First, the scheduling window module defines the number of available processors according to the energy case. Second, the scheduling algorithm uses a backfilling algorithm to place the jobs in the available processors. This scheduler receives a job priority, considering it in the decision process. Finally, they do a job power profiling, providing data to create a job power model.

In [55], the authors defined an online energy-aware scheduling algorithm using deep reinforcement learning. The main idea is to use the grid power when the carbon emission rate is lower. They used a DAG to define the different tasks of a batch job. The DAG indicates when a task can start (e.g., task 2 can run after task 1 is finished). A typical reinforcement learning algorithm has three key elements: state, actions, and reward. The state in this article is given by server usage, task queue, electricity price, and emission rate. The task queue contains only the tasks ready to run, considering the job's DAG. The actions are the tuple of task and server, considering the feasibility of this tuple (if a server can not receive the task, it is not a feasible action). Finally, the reward is carbon, cost, and QoS. They do not introduce server decisions, so the servers are always available.

2.5.3 Mixed decisions

The only work that mixed offline and online decisions is [92]. In the offline part, the authors used renewable energy predictions to define the percentage of resources available. They use these predictions, energy storage, and brown energy to fix the number of resources. After that, online makes the scheduling decisions considering the offline server configuration. They created a deep reinforcement learning algorithm to define the jobs to run. The state is composed of tuples containing both the resource availability and the array of job metadata. The job metadata includes the price that the user is willing to pay, QoS, expected finish time, duration, and resource requirements. The actions are which job run, suspend a job, or do nothing. So, they can suspend a job, placing a job with a higher price first. This suspended job will run later. Finally, the reward is the total value obtained by running the jobs respecting the QoS. The authors indicate that power decisions would transform their problem into a multi-criteria optimization problem. These power decisions include changing battery usage and selecting power sources. They claimed that this multi-criteria optimization is future work. They evaluated the impact of the power supply intermittence on the algorithm, but DRL does not consider it in the model.

2.5.4 Discussion and Classification of the Literature

Article	Year	Objective	Electrical infrastructure	Offline decisions	Online decisions	Method
Aksanli et al. [9]	2011	Maximize green energy usage	Solar panels, wind turbines, and grid	-	Server and Scheduling	Heuristic
Goiri et al. [31]	2015	Maximize green energy usage and reduce grid energy cost	Solar panels and grid	-	Server and Scheduling	Heuristic
Gu et al. [34]	2015	Minimize carbon emissions	Solar panels, wind turbines, and grid	Server, Scheduling, and Power	-	Exact algorithm
Li et al. [53]	2016	Balance QoS, battery life span, and average backup time	Wind turbines, batteries, and grid	-	Server, Power	Heuristic
Kassab et al. [46]	2017	Minimize makespan and flowtime	Solar panels and wind turbines	Scheduling	-	Heuristic
Li et al. [54]	2017	Maximize green energy usage	Solar, batteries, and grid	-	Server, Scheduling, and Power	Heuristic
Kassab et al. [47]	2018	Minimize makespan and flowtime	Solar panels and wind turbines	Scheduling	-	Metaheuristic
Grange et al. [32]	2018	Minimize grid energy and respect QoS	Solar and grid	-	Server and Scheduling	Heuristic

Article	Year	Objective	Electrical infrastructure	Offline decisions	Online decisions	Method
Hu et al. [41]	2018	Minimize makespan under energy constraints	-	Scheduling	-	Heuristic
Lu et al. [56]	2018	Minimize energy cost	Solar panels and grid	Scheduling and Power	-	Exact algorithm
Caux et al. [17]	2018	Maximize QoS under power constraints	Solar panels and wind turbines	Server and Scheduling	-	Metaheuristic and heuristic
Caux et al. [18]	2019	Maximize profit under power constraints	Solar panels and wind turbines	Server and Scheduling	-	Heuristic
Haddad et al. [35]	2019	Match power demand and production	Solar panels, wind turbines, batteries, and hydrogen	Power	-	Exact algorithm
Gao et al. [29]	2020	Match power demand and production minimizing QoS violations	Solar panels, wind turbines, and grid	Power	-	Exact algorithm and machine learning
Haghshenas et al. [36]	2020	Minimize energy cost	Solar panels, batteries, diesel generator, and grid	-	Scheduling, and Power	Heuristic
Nayak et al. [66]	2021	Minimize makespan, energy consumption, and overall cost, and maximize renewable usage	Not specified (renewable and non-renewable without battery)	-	Scheduling	Heuristic

Article	Year	Objective	Electrical infrastructure	Offline decisions	Online decisions	Method
He et al. [38]	2021	Minimize energy cost	Solar panels, wind turbines, and grid	-	Power	Heuristic
Peng et al. [72]	2022	Minimize energy cost	Solar panels, wind turbines, batteries, diesel generator, and grid	-	Server, Scheduling, and Power	Heuristic
Wiesner et al. [96]	2022	Maximize renewable excess energy usage	Solar panels and wind turbines	Scheduling	-	Heuristic
Yuan et al. [99]	2022	Minimize energy cost	Solar panels, wind turbines, batteries, and grid	Power	-	Exact algorithm
Liu et al. [55]	2023	Minimize the energy consumption cost and carbon footprint	Grid	-	Scheduling	Machine learning
Venkataswamy et al. [92]	2023	Maximize job value (revenue)	Solar panels, wind turbines, batteries, and grid	Server	Scheduling	Machine learning

Chapter 3

Modelling, Data, and Simulation

Contents

3.1 Model 34

3.2 Data 34

3.3 Simulation 34

3.4 Conclusion 34

3.1 Model

3.1.1 Offline Decision Modules

Power Decision Module

IT Decision Module

3.1.2 Offline Plan

3.1.3 Online Decision Modules

Job scheduling

Modifying Power Plan

Modifying IT Plan

3.2 Data

3.2.1 Workload Trace

3.2.2 Weather Trace

3.2.3 Platform Configuration

3.3 Simulation

3.3.1 Simulator

3.3.2 Metrics

3.3.3 Datazero2 Middleware

3.4 Conclusion

Chapter 4

Introducing Power Compensations

Contents

4.1 Introduction 35
4.2 Model 35
4.3 Heuristics 35
4.4 Results Evaluation 35
4.5 Conclusion 35

4.1 Introduction

4.2 Model

4.3 Heuristics

4.4 Results Evaluation

4.5 Conclusion

Chapter 5

Learning Power Compensations

Contents

5.1 Introduction 37
5.2 Algorithms 37
5.3 States 37
5.4 Actions 37
5.5 Rewards 37
5.6 Results Evaluation 37
5.7 Conclusion 37

5.1 Introduction

5.2 Algorithms

5.2.1 Random

5.2.2 Q-Learning approach

5.2.3 Contextual Multi-Armed Bandit approach

5.3 States

5.4 Actions

5.5 Rewards

5.6 Results Evaluation

5.7 Conclusion

Chapter 6

Adding Battery Awareness in EASY Backfilling

Contents

6.1 Introduction 39
6.2 Model 39
6.3 Heuristic 39
6.4 Conclusion 39

6.1 Introduction

6.2 Model

6.3 Heuristic

6.3.1 Predictions

6.3.2 Job Scheduling

6.3.3 Power compensation

6.4 Conclusion

Chapter 7

Conclusion and Perspectives

7.1 Conclusion

7.2 Perspectives

Bibliography

[1] Klimastatus og -fremskrivning 2023. https://ens.dk/service/fremskrivninger-analyser-modeller/klimastatus-og-fremskrivning-2023. Accessed: 2023-06-07.

[2] Electricity domestic consumption. https://yearbook.enerdata.net/electricity/electricity-domestic-consumption-data.html. Accessed: 2023-06-07.

[3] Measuring digital development - facts and figures 2022. https://www.itu.int/itu-d/reports/statistics/facts-figures-2022/. Accessed: 2023-06-07.

[4] Data centres metered electricity consumption 2021. https://www.cso.ie/en/releasesandpublications/ep/p-dcmec/datacentresmeteredelectricityconsumption2021/keyfindings/. Accessed: 2023-06-07.

[5] What is renewable energy? https://www.un.org/en/climatechange/what-is-renewable-energy. Accessed: 2023-06-08.

[6] What is a data center? https://www.ibm.com/topics/data-centers. Accessed: 2023-06-07.

[7] Amazon sets a new record for most renewable energy purchased by a single company. https://www.aboutamazon.eu/news/sustainability/amazon-sets-a-new-record-for-most-renewable-energy-purchased-by-a-single-company, 2023. Accessed: 2023-06-08.

[8] Pragati Agrawal and Shrisha Rao. Energy-efficient scheduling: classification, bounds, and algorithms. *Sādhanā*, 46(1):46, 2021.

[9] Baris Aksanli, Jagannathan Venkatesh, Liuyi Zhang, and Tajana Rosing. Utilizing green energy prediction to schedule mixed batch and service jobs in data centers. In *Proceedings of the 4th workshop on power-aware computing and systems*, pages 1–5, 2011.

[10] Nana Yaw Amponsah, Mads Troldborg, Bethany Kington, Inge Aalders, and Rupert Lloyd Hough. Greenhouse gas emissions from renewable energy sources: A review of lifecycle considerations. *Renewable and Sustainable Energy Reviews*, 39: 461–475, 2014.

[11] Anders SG Andrae and Tomas Edler. On global electricity usage of communication technology: trends to 2030. *Challenges*, 6(1):117–157, 2015.

[12] Chad Augustine, Richard Bain, Jamie Chapman, Paul Denholm, Easan Drury, Douglas G Hall, Eric Lantz, Robert Margolis, Robert Thresher, Debra Sandor, et al. Renewable electricity futures study. volume 2. renewable electricity generation and storage technologies. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States), 2012.

[13] Lotfi Belkhir and Ahmed Elmeligi. Assessing ict global emissions footprint: Trends to 2040 & recommendations. *Journal of cleaner production*, 177:448–463, 2018.

[14] Piotr Bojek. Renewables - energy system overview. Technical report, International Energy Agency, Paris, 2022.

[15] Ilhem Boussaïd, Julien Lepagnot, and Patrick Siarry. A survey on optimization metaheuristics. *Information sciences*, 237:82–117, 2013.

[16] Mary Branscombe. Google's solar deal for nevada data center would be largest of its kind. *Informa PLC, London*, 2020.

[17] Stephane Caux, Paul Renaud-Goud, Gustavo Rostirolla, and Patricia Stolf. It optimization for datacenters under renewable power constraint. In *European Conference on Parallel Processing*, pages 339–351. Springer, 2018.

[18] Stephane Caux, Paul Renaud-Goud, Gustavo Rostirolla, and Patricia Stolf. Phase-based tasks scheduling in data centers powered exclusively by renewable energy. In *2019 31st International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*, pages 136–143. IEEE, 2019.

[19] U Cisco. Cisco annual internet report (2018–2023) white paper, 2020.

[20] Climate Action Tracker. Massive gas expansion risks overtaking positive climate policies. *Warming Projections Global Update, November*, 2022.

[21] Climate Action Tracker. 2100 warming projections: Emissions and expected warming based on pledges and current policies. *Warming Projections Global Update, November*, 2022. Available at: `https://climateactiontracker.org/global/temperatures/`.

[22] Georges Da Costa, Léo Grange, and Inès De Courchelle. Modeling, classifying and generating large-scale google-like workload. *Sustainable Computing: Informatics and Systems*, 19:305–314, 2018.

[23] Miyuru Dayarathna, Yonggang Wen, and Rui Fan. Data center energy consumption modeling: A survey. *IEEE Communications Surveys & Tutorials*, 18(1):732–794, 2015.

[24] Igor Fontana de Nardin, Patricia Stolf, and Stephane Caux. Analyzing power decisions in data center powered by renewable sources. In *2022 IEEE 34th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*, pages 305–314. IEEE, 2022.

[25] Igor Fontana de Nardin, Patricia Stolf, and Stephane Caux. Mixing offline and online electrical decisions in data centers powered by renewable sources. In *IECON 2022–48th Annual Conference of the IEEE Industrial Electronics Society*, pages 1–6. IEEE, 2022.

[26] Weiqiang Dong, Yanjun Li, and Ji Xiang. Optimal sizing of a stand-alone hybrid power system based on battery/hydrogen with an improved ant colony optimization. *Energies*, 9(10):785, 2016.

[27] Dror G Feitelson, Dan Tsafrir, and David Krakov. Experience with using the parallel workloads archive. *Journal of Parallel and Distributed Computing*, 74(10):2967–2982, 2014.

[28] Charlotte Freitag, Mike Berners-Lee, Kelly Widdicks, Bran Knowles, Gordon Blair, and Adrian Friday. The climate impact of ict: A review of estimates, trends and regulations, 2021.

[29] Jiechao Gao, Haoyu Wang, and Haiying Shen. Smartly handling renewable energy instability in supporting a cloud datacenter. In *2020 IEEE international parallel and distributed processing symposium (IPDPS)*, pages 769–778. IEEE, 2020.

[30] Raquel S Garcia and Daniel Weisser. A wind–diesel system with hydrogen storage: Joint optimisation of design and dispatch. *Renewable energy*, 31(14):2296–2320, 2006.

[31] Íñigo Goiri, Md E Haque, Kien Le, Ryan Beauchea, Thu D Nguyen, Jordi Guitart, Jordi Torres, and Ricardo Bianchini. Matching renewable energy supply and demand in green datacenters. *Ad Hoc Networks*, 25:520–534, 2015.

[32] Léo Grange, Georges Da Costa, and Patricia Stolf. Green it scheduling for data center powered with renewable energy. *Future Generation Computer Systems*, 86: 99–120, 2018.

[33] Robert Gross, Matthew Leach, and Ausilio Bauen. Progress in renewable energy. *Environment international*, 29(1):105–122, 2003.

[34] Chonglin Gu, Chunyan Liu, Jiangtao Zhang, Hejiao Huang, and Xiaohua Jia. Green scheduling for cloud data centers using renewable resources. In *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 354–359. IEEE, 2015.

[35] Marwa Haddad, Jean Marc Nicod, Christophe Varnier, and Marie-Cécile Peéra. Mixed integer linear programming approach to optimize the hybrid renewable energy system management for supplying a stand-alone data center. In *2019 Tenth international green and sustainable computing conference (IGSC)*, pages 1–8. IEEE, 2019.

[36] Kawsar Haghshenas, Somayye Taheri, Maziar Goudarzi, and Siamak Mohammadi. Infrastructure aware heterogeneous-workloads scheduling for data center energy cost minimization. *IEEE Transactions on Cloud Computing*, 2020.

[37] Ali Hammadi and Lotfi Mhamdi. A survey on architectures and energy efficiency in data center networks. *Computer Communications*, 40:1–21, 2014.

[38] Huaiwen He, Hong Shen, Qing Hao, and Hui Tian. Online delay-guaranteed workload scheduling to minimize power cost in cloud data centers using renewable energy. *Journal of Parallel and Distributed Computing*, 159:51–64, 2022.

[39] Franz Christian Heinrich, Tom Cornebize, Augustin Degomme, Arnaud Legrand, Alexandra Carpen-Amarie, Sascha Hunold, Anne-Cécile Orgerie, and Martin Quinson. Predicting the energy-consumption of mpi applications at scale using only a single node. In *2017 IEEE international conference on cluster computing (CLUSTER)*, pages 92–102. IEEE, 2017.

[40] John Houghton. Global warming. *Reports on progress in physics*, 68(6):1343, 2005.

[41] Fengsong Hu, Xiajie Quan, and Can Lu. A schedule method for parallel applications on heterogeneous distributed systems with energy consumption constraint. In *Proceedings of the 3rd International Conference on Multimedia Systems and Signal Processing*, pages 134–141, 2018.

[42] IPCC Climate Change. A threat to human wellbeing and health of the planet. *Taking Action Now Can Secure our Future*, 2022.

[43] Leila Ismail and Huned Materwala. Computing server power modeling in a data center: Survey, taxonomy, and performance evaluation. *ACM Computing Surveys (CSUR)*, 53(3):1–34, 2020.

[44] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[45] George Kamiya. Data centres and data transmission networks. Technical report, International Energy Agency, Paris, 2022.

[46] Ayham Kassab, Jean-Marc Nicod, Laurent Philippe, and Veronika Rehn-Sonigo. Scheduling independent tasks in parallel under power constraints. In *2017 46th International Conference on Parallel Processing (ICPP)*, pages 543–552. IEEE, 2017.

[47] Ayham Kassab, Jean-Marc Nicod, Laurent Philippe, and Veronika Rehn-Sonigo. Assessing the use of genetic algorithms to schedule independent tasks under power constraints. In *2018 International Conference on High Performance Computing & Simulation (HPCS)*, pages 252–259. IEEE, 2018.

[48] Ayham Kassab, Jean-Marc Nicod, Laurent Philippe, and Veronika Rehn-Sonigo. Green power constrained scheduling for sequential independent tasks on identical parallel machines. In *2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, pages 132–139. IEEE, 2019.

[49] Md Anit Khan, Andrew P Paplinski, Abdul Malik Khan, Manzur Murshed, and Rajkumar Buyya. Exploiting user provided information in dynamic consolidation of virtual machines to minimize energy consumption of cloud data centers. In *2018 Third International Conference on Fog and Mobile Edge Computing (FMEC)*, pages 105–114. IEEE, 2018.

[50] Dalibor Klusáček, Šimon Tóth, and Gabriela Podolníková. Real-life experience with major reconfiguration of job scheduling system. In *Job Scheduling Strategies for Parallel Processing: 19th and 20th International Workshops, JSSPP 2015, Hyderabad, India, May 26, 2015 and JSSPP 2016, Chicago, IL, USA, May 27, 2016, Revised Selected Papers 19*, pages 83–101. Springer, 2017.

[51] Pierre L Kunsch and Jean Friesewinkel. Nuclear energy policy in belgium after fukushima. *Energy policy*, 66:462–474, 2014.

[52] Timothy M Lenton, Chi Xu, Jesse F Abrams, Ashish Ghadiali, Sina Loriani, Boris Sakschewski, Caroline Zimm, Kristie L Ebi, Robert R Dunn, Jens-Christian Svenning, et al. Quantifying the human cost of global warming. *Nature Sustainability*, pages 1–11, 2023.

[53] Chao Li, Rui Wang, Depei Qian, and Tao Li. Managing server clusters on renewable energy mix. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 11 (1):1–24, 2016.

[54] Yunbo Li, Anne-Cécile Orgerie, and Jean-Marc Menaud. Balancing the use of batteries and opportunistic scheduling policies for maximizing renewable energy consumption in a cloud data center. In *2017 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, pages 408–415. IEEE, 2017.

[55] Wenyu Liu, Yuejun Yan, Yimeng Sun, Hongju Mao, Ming Cheng, Peng Wang, and Zhaohao Ding. Online job scheduling scheme for low-carbon data center operation: An information and energy nexus perspective. *Applied Energy*, 338:120918, 2023.

[56] Yiwen Lu, Ran Wang, Ping Wang, Yue Cao, Jie Hao, and Kun Zhu. Energy-efficient task scheduling for data centers with unstable renewable energy: A robust optimization approach. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 455–462. IEEE, 2018.

[57] Yiwen Lu, Ran Wang, Ping Wang, Yue Cao, Jie Hao, and Kun Zhu. Energy-Efficient Task Scheduling for Data Centers with Unstable Renewable Energy: A Robust Optimization Approach. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 455–462, July 2018. doi: 10.1109/Cybermatics_2018.2018.00101.

[58] Akbar Maleki and Fathollah Pourfayaz. Optimal sizing of autonomous hybrid photovoltaic/wind/battery power system with lpsp technology by using evolutionary algorithms. *Solar Energy*, 115:471–483, 2015.

[59] Jens Malmodin. The ict sector's carbon footprint, 2020. URL `https://spark.adobe.com/page/dey6WTCZ5JKPu/`.

[60] Jens Malmodin and Dag Lundén. The energy and carbon footprint of the global ict and e&m sectors 2010–2015. *Sustainability*, 10(9):3027, 2018.

[61] Eric Masanet, Arman Shehabi, Nuoa Lei, Sarah Smith, and Jonathan Koomey. Recalibrating global data center energy-use estimates. *Science*, 367(6481):984–986, 2020. ISSN 0036-8075. doi: 10.1126/science.aba3758. URL `https://science.sciencemag.org/content/367/6481/984`.

[62] Mohammad Masdari and Afsane Khoshnevis. A survey and classification of the workload forecasting methods in cloud computing. *Cluster Computing*, 23(4):2399–2424, 2020.

[63] Valérie Masson-Delmotte, Panmao Zhai, Hans-Otto Pörtner, Debra Roberts, Jim Skea, Priyadarshi R Shukla, Anna Pirani, Wilfran Moufouma-Okia, Clotilde Péan, Roz Pidcock, et al. Global warming of 1.5 c. *An IPCC Special Report on the impacts of global warming of*, 1(5):43–50, 2018.

[64] Gordon E Moore et al. Cramming more components onto integrated circuits, 1965.

[65] United Nations. The Paris Agreement. URL `https://www.un.org/en/climatechange/paris-agreement`. Publisher: United Nations.

[66] Sanjib Kumar Nayak, Sanjaya Kumar Panda, Satyabrata Das, and Sohan Kumar Pande. An efficient renewable energy-based scheduling algorithm for cloud computing. In *International Conference on Distributed Computing and Internet Technology*, pages 81–97. Springer, 2021.

[67] AG Olabi and Mohammad Ali Abdelkareem. Renewable energy and climate change. *Renewable and Sustainable Energy Reviews*, 158:112111, 2022.

[68] Anne-Cecile Orgerie, Laurent Lefevre, and Jean-Patrick Gelas. Demystifying energy consumption in grids and clouds. In *International Conference on Green Computing*, pages 335–342. IEEE, 2010.

[69] Anne-Cecile Orgerie, Marcos Dias de Assuncao, and Laurent Lefevre. A survey on techniques for improving the energy efficiency of large-scale distributed systems. *ACM Computing Surveys (CSUR)*, 46(4):1–31, 2014.

[70] N L Panwar, S C Kaushik, and Surendra Kothari. Role of renewable energy sources in environmental protection: A review. *Renewable and sustainable energy reviews*, 15(3):1513–1524, 2011.

[71] Michael K Patterson. The effect of data center temperature on energy efficiency. In *2008 11th Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, pages 1167–1174. IEEE, 2008.

[72] Xiaopu Peng, Tathagata Bhattacharya, Jianzhou Mao, Ting Cao, Chao Jiang, and Xiao Qin. Energy-efficient management of data centers using a renewable-aware scheduler. In *2022 IEEE International Conference on Networking, Architecture and Storage (NAS)*, pages 1–8. IEEE, 2022.

[73] Ignacio J Perez-Arriaga. Managing large scale penetration of intermittent renewables. In *MITEI Symposium on Managing Large-Scale Penetration of Intermittent Renewables, Cambridge/USA*, volume 20, page 2011, 2011.

[74] Stefan Pfenninger and Iain Staffell. Long-term patterns of european pv output using 30 years of validated hourly reanalysis and satellite data. *Energy*, 114:1251–1265, 2016.

[75] Jean-Marc Pierson, Gwilherm Baudic, Stéphane Caux, Berk Celik, Georges Da Costa, Léo Grange, Marwa Haddad, Jerome Lecuivre, Jean-Marc Nicod, Laurent Philippe, Veronika Rehn-Sonigo, Robin Roche, Gustavo Rostirolla, Amal Sayah, Patricia Stolf, Minh-Thuyen Thi, and Christophe Varnier. DATAZERO: DATAcenter with Zero Emission and RObust management using renewable energy. *IEEE Access*, 7:(on line), juillet 2019. URL `http://doi.org/10.1109/ACCESS.2019.2930368`.

[76] Thomas Pregger, Daniela Graf, Wolfram Krewitt, Christian Sattler, Martin Roeb, and Stephan Möller. Prospects of solar thermal hydrogen production processes. *International journal of hydrogen energy*, 34(10):4256–4267, 2009.

[77] Issam Raïs, Anne-Cécile Orgerie, Martin Quinson, and Laurent Lefèvre. Quantifying the impact of shutdown techniques for energy-efficient data centers. *Concurrency and Computation: Practice and Experience*, 30(17):e4471, 2018.

[78] Charles Reiss, John Wilkes, and Joseph L Hellerstein. Google cluster-usage traces: format+ schema. *Google Inc., White Paper*, 1:1–14, 2011.

[79] Chuangang Ren, Di Wang, Bhuvan Urgaonkar, and Anand Sivasubramaniam. Carbon-aware energy capacity planning for datacenters. In *2012 IEEE 20th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pages 391–400. IEEE, 2012.

[80] Yves Robert and Frédéric Vivien. *Introduction to scheduling*. CRC Press, 2009.

[81] Gustavo Rostirolla, Léo Grange, T Minh-Thuyen, Patricia Stolf, Jean-Marc Pierson, Georges Da Costa, Gwilherm Baudic, Marwa Haddad, Ayham Kassab, Jean-Marc Nicod, et al. A survey of challenges and solutions for the integration of renewable energy in datacenters. *Renewable and Sustainable Energy Reviews*, 155:111787, 2022.

[82] Navin Sharma, Sean Barker, David Irwin, and Prashant Shenoy. Blink: managing server clusters on intermittent power. In *Proceedings of the sixteenth international conference on Architectural support for programming languages and operating systems*, pages 185–198, 2011.

[83] Rahul Sharma and Diksha Singh. A review of wind power and wind speed forecasting. *Journal of Engineering Research and Application*, 8(7):1–9, 2018.

[84] Sunanda Sinha and SS Chandel. Review of recent trends in optimization techniques for solar photovoltaic–wind based hybrid energy systems. *Renewable and Sustainable Energy Reviews*, 50:755–769, 2015.

[85] Saurabh S Soman, Hamidreza Zareipour, Om Malik, and Paras Mandal. A review of wind power and wind speed forecasting methods with different time horizons. In *North American power symposium 2010*, pages 1–8. IEEE, 2010.

[86] Srividya Srinivasan, Rajkumar Kettimuthu, Vijay Subramani, and Ponnuswamy Sadayappan. Characterization of backfilling strategies for parallel job scheduling. In *Proceedings. International Conference on Parallel Processing Workshop*, pages 514–519. IEEE, 2002.

[87] Edward Baleke Ssekulima, Muhammad Bashar Anwar, Amer Al Hinai, and Mohamed Shawky El Moursi. Wind speed and solar irradiance forecasting techniques for enhanced renewable energy integration with the grid: a review. *IET Renewable Power Generation*, 10(7):885–989, 2016.

[88] Shinichiro Takizawa and Ryousei Takano. Effect of an incentive implementation for specifying accurate walltime in job scheduling. In *Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region*, pages 169–178, 2020.

[89] Aidan Tuohy, John Zack, Sue Ellen Haupt, Justin Sharp, Mark Ahlstrom, Skip Dise, Eric Grimit, Corinna Mohrlen, Matthias Lange, Mayte Garcia Casado, et al. Solar forecasting: methods, challenges, and performance. *IEEE Power and Energy Magazine*, 13(6):50–59, 2015.

[90] UNESCO. Guide to measuring information and communication technologies (ict) in education, 2009.

[91] Avneesh Vashistha and Pushpneel Verma. A literature review and taxonomy on workload prediction in cloud data center. In *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 415–420. IEEE, 2020.

[92] Vanamala Venkataswamy, Jake Grigsby, Andrew Grimshaw, and Yanjun Qi. Rare: Renewable energy aware resource management in datacenters. In *Job Scheduling Strategies for Parallel Processing: 25th International Workshop, JSSPP 2022, Virtual Event, June 3, 2022, Revised Selected Papers*, pages 108–130. Springer, 2023.

[93] Di Wang, Chuangang Ren, Anand Sivasubramaniam, Bhuvan Urgaonkar, and Hosam Fathy. Energy storage in datacenters: what, where, and how much? In *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE joint international conference on Measurement and Modeling of Computer Systems*, pages 187–198, 2012.

[94] Kangjin Wang, Ying Li, Cheng Wang, Tong Jia, Kingsum Chow, Yang Wen, Yaoyong Dou, Guoyao Xu, Chuanjia Hou, Jie Yao, et al. Characterizing job microarchitectural profiles at scale: Dataset and analysis. In *Proceedings of the 51st International Conference on Parallel Processing*, pages 1–11, 2022.

[95] Tim Wheeler and Joachim von Braun. Climate change impacts on global food security. *Science*, 341(6145):508–513, 2013. doi: 10.1126/science.1239402. URL `https://www.science.org/doi/abs/10.1126/science.1239402`.

[96] Philipp Wiesner, Dominik Scheinert, Thorsten Wittkopp, Lauritz Thamsen, and Odej Kao. Cucumber: Renewable-aware admission control for delay-tolerant cloud and edge workloads. In *Euro-Par 2022: Parallel Processing: 28th International Conference on Parallel and Distributed Computing, Glasgow, UK, August 22–26, 2022, Proceedings*, pages 218–232. Springer, 2022.

[97] Bolun Xu, Alexandre Oudalov, Andreas Ulbig, Göran Andersson, and Daniel S Kirschen. Modeling of lithium-ion battery degradation for cell life assessment. *IEEE Transactions on Smart Grid*, 9(2):1131–1140, 2016.

[98] Ahmet Yilanci, Ibrahim Dincer, and Hasan K Ozturk. A review on solar-hydrogen/fuel cell hybrid energy systems for stationary applications. *Progress in energy and combustion science*, 35(3):231–244, 2009.

[99] Jindou Yuan, Wenhan Zhang, Ying Zhou, Songsong Chen, and Ciwei Gao. Optimal scheduling of data centers considering renewable energy consumption and temporalspatial load characteristics. In *2022 Power System and Green Energy Conference (PSGEC)*, pages 283–288. IEEE, 2022.

[100] Qingxia Zhang, Zihao Meng, Xianwen Hong, Yuhao Zhan, Jia Liu, Jiabao Dong, Tian Bai, Junyu Niu, and M Jamal Deen. A survey on data center cooling systems: Technology, power consumption modeling and control strategy optimization. *Journal of Systems Architecture*, 119:102253, 2021.