electronics

MDPI

Article

Energy Efficiency Evaluation of Artificial Intelligence Algorithms †

Kalin Penev [1,*], Alexander Gegov [2,3], Olufemi Isiaq [4] and Raheleh Jafari [5]

[1] Department of Science and Engineering, Solent University, Southampton SO14 0YN, UK
[2] School of Computing, University of Portsmouth, Portsmouth PO1 3HE, UK; alexander.gegov@port.ac.uk
[3] English Faculty of Engineering, Technical University of Sofia, 1756 Sofia, Bulgaria
[4] Creative Computing Institute, University of the Arts London, London WC1V 7EY, UK; f.isiaq@arts.ac.uk
[5] School of Design, University of Leeds, Leeds LS2 9JT, UK; r.jafari@leeds.ac.uk
* Correspondence: kalin.penev@solent.ac.uk; Tel.: +44-(0)23-8201-3640
† In memory of the 100th anniversary of the birth of Ivan Georgiev Chouroulinkov, one of the great scientists of the 20th century.

Abstract: This article advances the discourse on sustainable and energy-efficient software by examining the performance and energy efficiency of intelligent algorithms within the framework of green and sustainable computing. Building on previous research, it explores the theoretical implications of Bremermann's limit on efforts to enhance computer performance through more extensive methods. The study presents an empirical investigation into heuristic methods for search and optimisation, demonstrating the energy efficiency of various algorithms in both simple and complex tasks. It also identifies key factors influencing the energy consumption of algorithms and their potential impact on computational processes. Furthermore, the article discusses cognitive concepts and their interplay with computational intelligence, highlighting the role of cognition in the evolution of intelligent algorithms. The conclusion offers insights into the future directions of research in this area, emphasising the need for continued exploration of energy-efficient computing methodologies.

Keywords: green computing; software energy efficiency; sustainable and responsible artificial intelligence; Free Search; Bremermann's limit

Citation: Penev, K.; Gegov, A.; Isiaq, O.; Jafari, R. Energy Efficiency Evaluation of Artificial Intelligence Algorithms. *Electronics* **2024**, *13*, 3836. https://doi.org/10.3390/electronics13193836

Academic Editor: Maciej Ławryńczuk

Received: 24 June 2024
Revised: 19 September 2024
Accepted: 25 September 2024
Published: 28 September 2024

1. Introduction

The increasing enthusiasm for environmentally sustainable practices naturally extends to green computing, where software plays a pivotal role. Green computing encompasses a wide range of strategies, including optimising hardware and software design, reducing power consumption, employing renewable energy sources, enhancing software efficiency, virtualising servers, and managing electronic waste (e-waste) [1]. The goal of green computing is to improve the performance and energy efficiency of computational systems through both hardware and software optimisation.

Recent societal and environmental concerns have driven the focus towards Responsible Artificial Intelligence (RAI), which aims to develop energy-efficient intelligent software systems [2,3]. While Artificial Intelligence (AI) holds the potential to create a future where all of humanity can thrive, the energy consumption of Information Technologies (IT)—including portable devices, data centres, and cloud servers—has been escalating annually [4]. This surge in energy demand is reflected in global carbon emissions, as highlighted in recent global energy reviews [5,6].

The issue of computing energy efficiency requires a deeper examination. As noted in foundational research, there is a theoretical upper limit to the rate at which data processing can occur. This limit is applicable to all data processing systems, whether artificial or biological, and posits that "no data processing system, artificial or living, can process more than (2 times 10^{47}) bits per second per gram of its mass" [7]. The formulation of these computational constraints is grounded in fundamental physical principles: "The capacity of any closed information transmission or processing does not exceed ($Mc^{2/h}$ = ~(M/gram)10^{47}) bits per

second, where (M) is the mass of the system, (c) is the speed of light, and (h) is Planck's constant" [8].

More recent studies have refined this understanding, suggesting that Bremermann's limit, initially proposed in 1962, should be corrected to align with the principles of general relativity. The revised limit is expressed as $(c^5/Gh)^{1/2} = \sim 10^{43}$ bits per second, where (c) is the speed of light, (G) is the gravitational constant, and (h) is Planck's constant [9]. The existence of Bremermann's limit suggests that further improvements in computational performance will encounter insurmountable physical barriers. Early signs that computing systems are nearing this threshold include the growing need for cooling systems to dissipate heat, increased electricity consumption, and changes in environmental heat pollution that may have irreversible effects on the climate.

A potential solution to these limitations is to enhance energy efficiency in a manner similar to natural systems, focusing on both hardware and software optimisation. While there has been significant progress in the energy efficiency of hardware in large computational systems—including those used by major AI services such as OpenAI—with supercomputers improving their hardware energy efficiency by more than 200 times over the past 20 years based on the LINPAC Benchmark [10–13], such categorical advancements in software energy efficiency remain elusive.

This article, therefore, focuses on the critical need for improving software energy efficiency, particularly in the context of AI and intelligent algorithms, which often handle high volumes of uncertain, time-dependent data.

2. Survey of Related Literature

While hardware energy efficiency has consistently improved over time [14,15], concerns about the energy efficiency of software have only recently emerged, as evidenced by a growing number of publications on the topic [16]. Software sustainability encompasses a range of applications, including specific software products, online applications, and data processing systems. This involves minimising power consumption and optimising the entire software lifecycle, considering human, economic, and energy resources [16].

Given the increasing demand for portable devices such as smartphones, tablets, and laptops, significant research efforts have focused on reducing their energy consumption by enhancing software quality through techniques such as code refactoring, which restructures existing source code to be more energy efficient [17–19]. A more advanced approach involves assessing software sustainability holistically, considering factors such as efficiency, quality, and other critical properties [20–22]. This approach encourages software practitioners to prioritize sustainability during design and development [20], while offering systematic guidelines and frameworks that help professionals evaluate the sustainability impact of software [21,22].

Artificial Intelligence (AI) and intelligent systems, as sophisticated types of software, are fundamental to computing, cloud services, and data processing, influencing multiple aspects of life. Current research predominantly focuses on enhancing the accuracy and reliability of AI-based systems, which often requires vast datasets, large AI models, and resource-intensive infrastructures [23]. Recent studies have proposed a hypothesis suggesting that when developing "green" AI systems, architectural decisions' impact on energy efficiency must be better understood, managed, and reported to reduce computational power requirements [23].

Research also indicates that intelligent software development benefits from appropriate data abstractions, heuristic and metaheuristic algorithms, and the reduction of outdated limitations [24,25]. This allows for the adaptation of intelligent software to solve tasks with minimal computational resources, whether these tasks are relatively simple [24,26] or involve problems with a high number of parameters [27,28]. Consequently, there has been a shift towards developing sustainable, green AI-based software systems that utilize architecture-centric methods to model and develop energy-efficient AI systems [23]. How-

ever, these approaches often overlook essential properties of natural intelligence, such as cognition and adaptation, which are discussed further in this article.

Furthermore, a comprehensive study on design patterns for machine learning applications identifies 15 distinct patterns, such as solutions for real-time data processing and continuous reprocessing or storing data in a raw format [29]. However, these patterns are generally limited to analytical programming and fail to incorporate elements like abstraction and intuition, which are integral to natural intelligence. Evaluating AI systems' sustainability and quality also requires measuring and assessing software products and components' energy efficiency. Recent research efforts have focused on developing a Green Software Measurement Model to categorize existing measurement methods and create adapted methods for specific use cases, such as software types and system components [30]. This model has been adapted for the empirical research and evaluation of experimental software presented in this article.

Another aspect that requires further exploration is the impact of programming language choice on software energy efficiency. Recent studies have shown significant variations in energy consumption depending on the language and compiler used [31]. This highlights the need for more research into the influence of not only computer languages but also human languages on software energy consumption.

There are growing concerns regarding the current trajectory of AI, machine learning, and deep learning due to their exponentially increasing demand for data, training, and infrastructure [32,33]. These trends conflict with emerging regulations and requirements for efficiency and sustainability [34,35], as well as with the natural laws of selection [36]. Achieving harmony with these natural laws could enhance the sustainability of AI systems, prompting key questions, such as: How do biological systems manage data storage and transmission efficiently? Understanding these principles could inform the design of more sustainable AI.

In software engineering, especially for high-performance computing (HPC) systems, achieving a balance between energy efficiency and performance has become a critical non-functional requirement. Software developers must thoroughly understand both the problem domain and the target computer architecture, considering various programming models, languages, tools, and heterogeneous systems, which increases development complexity [37]. AI applications, in particular, demand high performance and energy efficiency, necessitating specialised knowledge from developers. Therefore, methodologies and tools that assist both specialised and general developers are crucial for optimising HPC systems. The time and energy consumption measurement approach discussed in this paper could be invaluable for evaluating intelligent computing and AI software systems.

In AI, computational intelligence, and software development, it is often observed that the same task can be accomplished using different resources and timeframes, similar to the behaviour of biological species. Examples in software include sorting algorithms [38] and adaptive heuristic algorithms in computational intelligence [39]. To enhance intelligent systems, genetic, swarm, evolutionary, heuristic, metaheuristic, and adaptive algorithms are promising. For instance, a study comparing over ten metaheuristic algorithms optimised by swarm intelligence for code smell detection demonstrated notable advancements in these algorithms' performance [40]. However, this research also highlighted common limitations among these metaheuristics, suggesting the need for further improvements.

The ultimate goal of AI is to develop technology that enables machines to operate in highly intelligent ways [41]. This objective drives the creation of new algorithms and large, high-quality datasets. However, it remains challenging for AI systems to address all potential real-world scenarios fully. Therefore, a critical question is how to harness these uncertainties to ensure socially responsible behaviour in AI algorithms [42]. Defining AI in a manner that aligns with social responsibility remains a significant challenge, and this study questions whether AI can be considered socially responsible based on its energy efficiency and sustainability, necessitating further comprehensive research [43].

3. Materials, Tools, and Methods

The selection of methodology and experimental settings for this study was guided by four key principles: minimising energy use, eliminating energy waste, thoroughly evaluating software processes, and avoiding specific settings that might unduly favour certain tasks.

To adhere to these principles, three algorithms and seven test problems were chosen, all of which have been previously studied with results documented in the literature. The selection of test problems was based on the following criteria:

1. The tests must involve problems with unknown optimal solutions.
2. The tests should be scalable to multidimensional formats.
3. The tests should feature heterogeneous landscapes.

The selected numerical tests meet these criteria, offering scalability and varying search spaces. Each test was configured with 100 parameters, and they include:

- Griewank test: A global optimisation problem with an optimal value of 0 [43].
- Michalewicz test: A global test with an unknown optimum that varies depending on the number of dimensions [44].
- Norwegian test: Another global test with an unknown optimum influenced by dimensionality [45].
- Rastrigin test: A global optimisation problem with an optimal value of 0 [46].
- Rosenbrock test: A smooth, flat test with a single solution and an optimal value of 0 [47].
- Schwefel test: A global optimisation problem with an optimal value of 0 [48].
- Step test: A test that introduces plateaus into the topology, which prevents reliance on local correlation in the search process. Its optimal value depends on the number of dimensions and may be unknown for various dimensions [49].

In alignment with the study's principles, three algorithms were selected:

- Particle Swarm Optimisation (PSO): A swarm-based algorithm for real-coded tasks over continuous spaces [50].
- Differential Evolution (DE): A heuristic algorithm designed for optimising nonlinear and non-differentiable functions in continuous spaces [51].
- Free Search (FS): An adaptive heuristic algorithm for search and optimisation within continuous spaces [52].

All algorithms were configured to operate on 10 candidate solutions, with a limit of 100,000 iterations over 320 sequential runs. The experiments aimed to measure both the processing time and energy consumption required to complete the specified number of iterations. Previous publications provide detailed evaluations of these algorithms in similar contexts [27,39,45,52].

The experiments were conducted on a computer system with the following specifications: an Intel XEON E5 1660 V2 processor overclocked to 4.750 GHz, operating in a 1 core—1 thread configuration with a maximum thermal design power (TDP) of 130 W. The system was equipped with a CPU water cooler, RAM set at 2000 MHz, an ASUS P9X79-E WS motherboard, and a SanDisk Extreme SSD SATA III solid-state drive. Each experiment was executed individually, with one algorithm applied to a single test function at a time to ensure accurate measurement of performance and energy consumption.

By adhering to these methodological principles and experimental setups, the study aims to provide robust, replicable, and meaningful insights into the efficiency of different optimisation algorithms under real-world conditions.

Methodology

This study adopts a modified version of the Green Software Measurement Model as proposed in prior literature [30]. The model was adapted to align with the specific context and requirements of this research by focusing on the following key parameters:

- Duration (min): Time taken for each experiment.
- Number of iterations (integer): The count of repeated cycles for each algorithm.
- Mean system power (*W*): Average power consumption of the entire system.
- System energy (W*h*): Total energy consumption over time.
- CPU usage (%): The percentage of CPU utilization during algorithm execution.
- CPU power (W): Power consumption specific to the CPU.
- CPU cores (1 core—1 thread): Number of active cores and threads during processing.

While the complete implementation of the Green Software Measurement Model as described in [30] is beyond the scope of this study, it remains a promising direction for future research, contingent on the availability of the required resources.

To measure power consumption, a digital power consumption energy meter was used to monitor the entire system's power usage [53]. The power consumption of each algorithm was calculated as the difference between the system's power level during the algorithm's execution and the baseline power level recorded when the system was in standby mode, with only the operating system components and tools for CPU parameter measurement (CPU-Z) [54] and CPU core temperature monitoring (Core Temp) [55] running. These monitoring tools were kept active throughout all experiments, running concurrently with the algorithms.

Each experiment was restricted to a maximum of 100,000 iterations, with the duration recorded manually at the start and the end time automatically logged as an attribute of the results file. This approach ensured that the time-tracking process did not interfere with the performance of the algorithms or the system configuration.

4. Results

The experimental results are summarised in Table 1, which presents the performance metrics of three optimisation algorithms: Particle Swarm Optimisation (PSO), Differential Evolution (DE), and Free Search (FS). The table outlines the time taken by each algorithm to complete the experiments, with the time recorded in the format of hours, minutes, and seconds (hh:mm:ss).

Table 1. Time for execution of 100-dimensional version of the tests.

Test	PSO	DE	FS
	Time	**Time**	**Time**
Griewank	01:45:00	00:41:00	00:14:00
Michalewicz	02:44:00	01:46:00	01:02:00
Norwegian	01:50:00	00:47:00	00:12:00
Rastrigin	01:46:00	00:45:00	00:11:00
Rosenbrock	01:39:00	00:40:00	00:05:00
Schwefel	02:44:00	01:03:00	00:27:00
Step	02:37:00	00:42:00	00:06:00

The mean system power consumption in standby mode, with the task monitor at 0% workload, was measured at 166 W on the socket. Under these conditions, the CPU power consumption for a single core with one thread was recorded at 33.4 W and 21.5 W, respectively.

At full workload capacity (100% workload) for all experiments reported in Table 1, the mean system power consumption increased to 185 W, with a variation of 3% throughout the execution period. This variation could potentially be attributed to changes in temperature or other environmental factors, which warrants further investigation. Additionally, under full workload conditions, the CPU power consumption for a single core with one thread was measured at 42.8 W and 30.4 W.

Presented in Table 1, data indicates variation of time for the execution per test and per algorithm.

The analysis of the experimental data reveals that the evaluation time per test is directly influenced by the complexity of the search space. More complex search spaces

require longer evaluation times. Additionally, the duration of the search process varies depending on the capabilities and characteristics of the algorithms used.

Figure 1 illustrates the time taken per test, while Figure 2 shows the time required by each algorithm to complete 100,000 iterations for a selected test case. Among the algorithms analysed, Particle Swarm Optimisation consistently required the most time across all tests. Differential Evolution exhibited a moderate range of time consumption, while the Free Search algorithm completed all tests the fastest. Notably, the results for the Schwefel and Step tests (Figure 2) suggest the presence of specific factors that may affect exploration time, indicating that certain features of these functions could be influencing the efficiency of the search process.

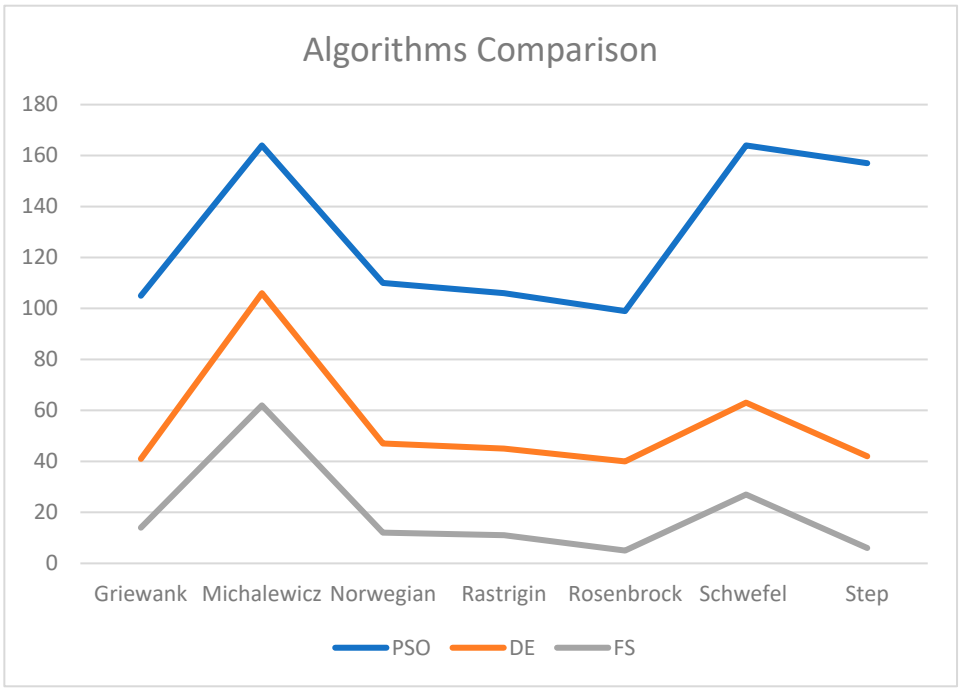

Figure 1. Tests time comparison per algorithm.

During the evaluation period, three distinct components can be considered:

- **Time for objective function evaluation**: This represents the duration required to understand and assess the search space.
- **Time for algorithm execution**: This refers to the time taken for the interpretation and assessment of the search space by the algorithm.
- **Time for algorithm decision making**: This is the duration needed for the algorithm to make decisions and select subsequent actions.

The energy consumption data presented in Table 2 is calculated based on the energy used by the algorithms. This is determined by the difference in power consumption between 100% workload during the experiments and 0% workload in standby mode, multiplied by the time taken to complete each experiment. Since different algorithms may take varying amounts of time to complete the same task, their energy consumption also differs accordingly. An analysis aimed at identifying systematic relationships between these components, summarised in Table 3, reveals only general qualitative differences.

The relative time differences (expressed as percentages in Table 3) generally suggest that the Differential Evolution (DE) algorithm is faster than Particle Swarm Optimisation (PSO) across all tests, while the Firefly Search (FS) algorithm is faster than both the Particle Swarm Optimisation (PSO) and DE algorithms. However, the magnitude of these differences varies significantly, and no precise systematic relationship can be identified

for each test or algorithm. A more detailed quantitative analysis could be the focus of future research.

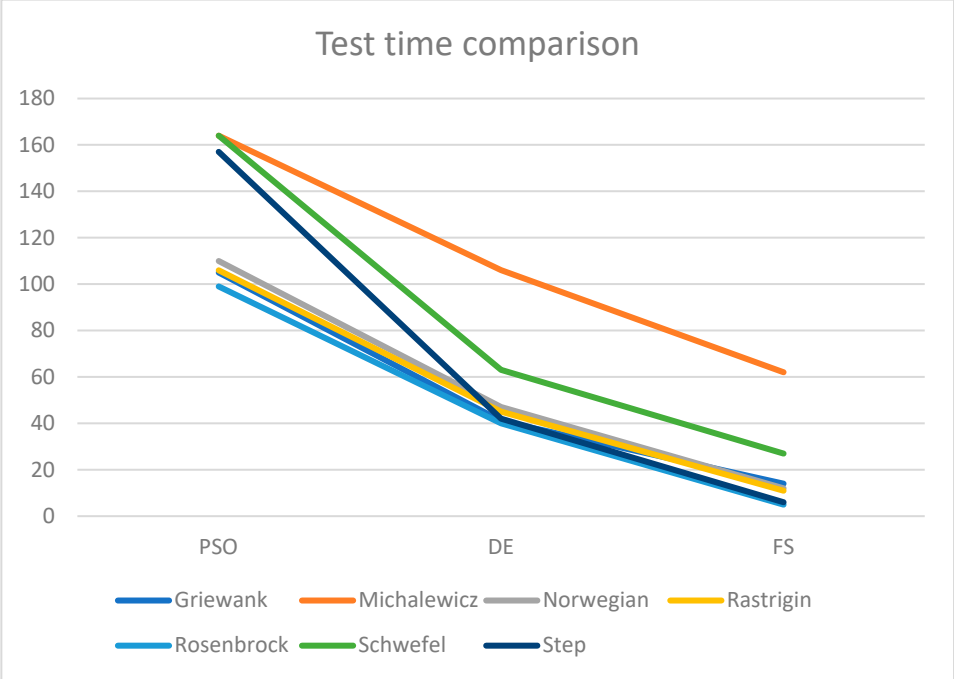

Figure 2. Algorithms time comparison per test.

Table 2. Energy use for execution of 100-dimensional version of the tests.

Test	PSO	DE	FS
	Wh	***Wh***	***Wh***
Griewank	33.25	12.98	4.43
Michalewicz	51.93	33.57	19.63
Norwegian	34.83	14.88	3.80
Rastrigin	33.57	14.25	3.48
Rosenbrock	31.35	12.67	1.58
Schwefel	51.93	19.95	8.55
Step	49.72	13.30	1.90

Table 3. Relative time difference per test in %.

Test	DE/PSO	FS/PSO	FS/DE
	%	**%**	**%**
Griewank	39%	13%	34%
Michalewicz	65%	38%	58%
Norwegian	43%	11%	26%
Rastrigin	42%	10%	24%
Rosenbrock	40%	5%	13%
Schwefel	38%	16%	43%
Step	27%	4%	14%

5. Discussion

This section critically examines the results of the study, interpreting them in light of previous research and exploring the role of intelligent algorithms in enhancing energy efficiency. The findings corroborate and, to some extent, clarify earlier studies [27] that investigate computational limitations, energy consumption, and processing time in intelligent

algorithms. While it is evident that variations in efficiency can be attributed to differences in software design, implementation, and execution, it is also essential to understand how different software engineering techniques can embody intelligent behaviour.

To ground this analysis, we first turn to epistemological frameworks [56,57]. Models such as Data–Information–Knowledge (DIK) [58] and Data–Information–Knowledge–Wisdom (DIKW) [59] provide valuable insights into how intelligent beings and systems perceive and interact with their environment. These models illustrate a hierarchical process wherein data is generated and pre-processed to abstract essential information, which can then be further refined into knowledge for future use. This hierarchical abstraction not only reduces the amount of data that needs to be stored but also accelerates the processing of familiar cases while enabling adaptation to new ones. Both factors significantly enhance the efficiency and sustainability of intelligent entities. Translating this process into intelligent computing software can thereby contribute to the overall sustainability of AI systems.

Among the various definitions of knowledge, the one most applicable to software design and implementation is "Knowledge is the perception of the agreement or disagreement of two ideas" [60–62]. This conceptualisation is crucial for understanding how cognitive processes can strengthen machine learning algorithms and improve the sustainability of intelligent systems. According to the literature, "Knowledge of the external world can be obtained either by intuition or by abstraction" [63]. Understanding these cognitive processes, particularly intuition and abstraction, is pivotal for advancing the process of machine learning and knowledge construction.

William of Ockham provides a useful distinction between intuitive and abstractive cognition [64]:

- *Intuitive cognition* involves the immediate apprehension that allows the intellect to make evident judgments about the existence or qualities of an object.
- *Abstractive cognition*, on the other hand, is an act of cognition where such judgments cannot be evidently made.

Applying these concepts to the "Blackbox" model facilitates the operation of adaptive heuristic algorithms such as Free Search, which can perform more efficiently across heterogeneous landscapes and tasks. Faster performance directly translates to better energy efficiency. For simple tasks, high computational intelligence and the competition between different algorithms and systems enhance software sustainability and energy efficiency. For more complex problems—such as those involving a search space exceeding $10^{1,000,000}$ (10 to the power of 1,000,000) possible locations, where the exploration time could approach infinity—adaptive intelligent behaviour becomes crucial in minimising both time and energy consumption. An example of this can be seen in the application of Free Search to optimise tasks involving 100,000 parameters, achieving notable efficiency gains [28].

Future research should focus on developing new models and software implementations that enhance machine learning, intelligent computing, environmental interaction, knowledge construction, and adaptive behaviour. These advancements are critical for creating more efficient and sustainable AI systems

6. Conclusions

This study contributes to the ongoing discourse on sustainable and energy-efficient software, specifically addressing the question: Can Artificial Intelligence (AI) be classified as socially responsible based on its energy efficiency and sustainability? While this question remains open and requires further comprehensive research, our findings provide a foundational perspective on the energy efficiency of computing systems, highlighting several key aspects:

1. The overall growth in energy consumption by computational systems poses significant challenges, especially considering the fundamental physical limitations that, if left unaddressed, could lead to global negative consequences.

2. Although there have been positive changes in hardware energy efficiency, the sustainability of software, particularly the energy efficiency of intelligent algorithms, plays a critical role.
3. Our empirical evaluation demonstrates the variation in time and energy consumption of intelligent, adaptive algorithms applied to heterogeneous numerical tests, revealing substantial differences in energy efficiency and speed when different algorithms are used for the same tasks.
4. The study identifies potential benefits of time- and energy-efficient software, underscoring the importance of optimising computational processes to reduce their environmental impact.
5. The discussion on the interrelationship between concepts, computational intelligence, and the role of cognition in advancing intelligent algorithms further elucidates the complexities involved in this area of study.

While the study provides valuable insights, it also raises several questions that remain unanswered and merit further investigation, such as the sustainability of other algorithms, the energy efficiency of algorithms when applied to real-world problems, and the broader contribution of intelligent computing to green computing. Future research should aim for more precise quantitative analyses, focusing on the evaluation and improvement of a wide range of software products and services to promote energy-efficient and sustainable computing.

By exploring these avenues, this study hopes to contribute to a deeper understanding of the potential for AI and other intelligent computing solutions to align with principles of social responsibility and environmental sustainability.

Author Contributions: Conceptualization, K.P. and A.G.; methodology, software, validation, K.P.; formal analysis, K.P. and O.I.; investigation, resources, data curation, K.P.; writing—original draft preparation, K.P.; writing—review and editing, K.P., A.G., O.I. and R.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

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

References

1. Paul, S.G.; Saha, A.; Arefin, M.S.; Bhuiyan, T.; Biswas, A.A.; Reza, A.W.; Alotaibi, N.M.; Alyami, S.A.; Moni, M.A. A Comprehensive Review of Green Computing: Past, Present, and Future Research. *IEEE Access* **2023**, *11*, 87445–87494. [CrossRef]
2. Cheng, L.; Varshney, K.R.; Liu, H. Socially responsible AI algorithms: Issues, purposes, and challenges. *J. Artif. Intell. Res.* **2021**, *71*, 1137–1181. [CrossRef]
3. Lee, S.U.; Fernando, N.; Lee, K.; Schneider, J. A survey of energy concerns for software engineering. *J. Syst. Softw.* **2024**, *210*, 111944. [CrossRef]
4. Naumann, S.; Dick, M.; Kern, E.; Johann, T. The GREENSOFT Model: A reference model for green and sustainable software and its engineering. *Sustain. Comput. Inform. Syst.* **2011**, *1*, 294–304. [CrossRef]
5. Raimi, D.; Zhu, Y.; Newell, R.G.; Prest, B.C. Global Energy Outlook 2024: Peaks or Plateaus? 2024. Available online: https://www.rff.org/publications/reports/global-energy-outlook-2024/ (accessed on 10 June 2024).
6. IEA. GlobalEnergyReview2021. Available online: https://www.iea.org/reports/global-energy-review-2021 (accessed on 10 June 2024).
7. Bremermann, H.J. Optimization through Evolution and Recombination. In *Self-Organizing Systems*; Yovits, M.C., Jacobim, G.T., Goldstein, G.D., Eds.; Spartan Books: Washington, DC, USA, 1962; pp. 93–106.
8. Bremermann, H.J. Quantum noise and information. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*; University of California Press: Berkeley, CA, USA, 1967; Volume 4, pp. 15–20.
9. Gorelik, G. Bremermann's Limit and cGh-physics. *arXiv* **2009**, arXiv:0910.3424. [CrossRef]
10. Top500A, Frontier-HPE Cray EX235A, AMD Optimized 3RD Generation EPYC 64C 2GHZ, AMD Instinct MI250X, SLINGSHOT-11. 2024. Available online: https://top500.org/system/180047/ (accessed on 15 May 2024).
11. Top500B, Bluegene/L-Eserver Blue Gene Solution. 2024. Available online: https://top500.org/system/174210/ (accessed on 15 May 2024).
12. Top500C, JEDI-Bullsequana XH3000, Grace Hopper Superchip 72C 3GHZ, NVIDIA GH200 Superchip, Quad-Rail NVIDIA Infiniband NDR200. 2024. Available online: https://top500.org/system/180269/ (accessed on 15 May 2024).

13. Dongarra, J. Frequently Asked Questions on the Linpack Benchmark and Top500. 2007. Available online: https://www.netlib.org/utk/people/JackDongarra/faq-linpack.html (accessed on 19 January 2024).
14. JVA Initiative Committee and Lowa State University, ATANASOFF BERRY COMPUTER. 2011. Available online: https://jva.cs.iastate.edu/operation.php (accessed on 12 June 2024).
15. Freiberger, P.A.; Swaine, M.R. "Atanasoff-Berry Computer". Encyclopedia Britannica. 20 March 2023. Available online: https://www.britannica.com/technology/Atanasoff-Berry-Computer (accessed on 11 June 2024).
16. Calero, C.; Mancebo, J.; Garcia, F.; Moraga, M.A.; Berna, J.A.G.; Fernandez-Aleman, J.L.; Toval, A. 5Ws of green and sustainable software. *Tsinghua Sci. Technol.* **2020**, *25*, 401–414. [CrossRef]
17. Gottschalk, M.; Jelschen, J.; Winter, A. Energy-Efficient Code by Refactoring. *Softwaretechnik-Trends* **2013**, *33*. Available online: https://api.semanticscholar.org/CorpusID:15332418 (accessed on 24 September 2024). [CrossRef]
18. Sanlıalp, İ.; Öztürk, M.M.; Yiğit, T. Energy Efficiency Analysis of Code Refactoring Techniques for Green and Sustainable Software in Portable Devices. *Electronics* **2022**, *11*, 442. [CrossRef]
19. Anwar, H.; Pfahl, D.; Srirama, S.N. Evaluating the impact of code smell refactoring on the energy consumption of Android applications. In Proceedings of the 2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Kallithea, Greece, 28–30 August 2019; pp. 82–86. [CrossRef]
20. Noman, H.; Mahoto, N.; Bhatti, S.; Rajab, A.; Shaikh, A. Towards sustainable software systems: A software sustainability analysis framework. *Inf. Softw. Technol.* **2024**, *169*, 107411. [CrossRef]
21. Heldal, R.; Nguyen, N.; Moreira, A.; Lago, P.; Duboc, L.; Betz, S.; Coroamă, V.C.; Penzenstadler, B.; Porras, J.; Capilla, R.; et al. Sustainability competencies and skills in software engineering: An industry perspective. *J. Syst. Softw.* **2024**, *211*, 111978. [CrossRef]
22. Venters, C.C.; Capilla, R.; Nakagawa, E.Y.; Betz, S.; Penzenstadler, B.; Crick, T.; Brooks, I. Sustainable software engineering: Reflections on advances in research and practice. *Inf. Softw. Technol.* **2023**, *164*, 107316. [CrossRef]
23. Martínez-Fernández, S.; Franch, X.; Durán, F. Towards green AI-based software systems: An architecture-centric approach (GAISSA). In Proceedings of the 2023 49th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Durres, Albania, 6–8 September 2023; pp. 432–439. [CrossRef]
24. Penev, K.; Littlefair, G. Free Search—A comparative analysis. *Inf. Sci.* **2005**, *172*, 173–193. [CrossRef]
25. Penev, K.; Gegov, A. (Eds.) *Free Search of Real Value or How to Make Computers Think*; BookSurge Publishing: Charleston, SC, USA, 2008; ISBN 978-0955894800.
26. Vasileva, V.; Penev, K. Free search of global value. In Proceedings of the 2012 6th IEEE International Conference Intelligent Systems, Sofia, Bulgaria, 6–8 September 2012; pp. 425–430. [CrossRef]
27. Penev, K. Free Search in Multidimensional Space M. In *Large-Scale Scientific Computing*; Lirkov, I., Margenov, S., Eds.; LSSC 2017. Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2018; Volume 10665, pp. 399–407. [CrossRef]
28. Penev, K. An Optimal Value for 100 000-Dimensional Michalewicz Test. 2022. Available online: https://pure.solent.ac.uk/files/33733992/100_000_dimensional_Michalewicz_test_2.pdf (accessed on 12 June 2024).
29. Washizaki, H.; Khomh, F.; Gueheneuc, Y.; Takeuchi, H.; Natori, N.; Doi, T.; Okuda, S. Software-Engineering Design Patterns for Machine Learning Applications. *Computer* **2022**, *55*, 30–39. [CrossRef]
30. Guldner, A.; Bender, R.; Calero, C.; Fernando, G.S.; Funke, M.; Gröger, J.; Hilty, L.M.; Hörnschemeyer, J.; Hoffmann, G.; Junger, D.; et al. Development and evaluation of a reference measurement model for assessing the resource and energy efficiency of software products and components—Green Software Measurement Model (GSMM). *Future Gener. Comput. Syst.* **2024**, *155*, 402–418. [CrossRef]
31. Koedijk, L.; Oprescu, A. Finding Significant Differences in the Energy Consumption when Comparing Programming Languages and Programs. In Proceedings of the 2022 International Conference on ICT for Sustainability (ICT4S), Plovdiv, Bulgaria, 13–17 June 2022; pp. 1–12. [CrossRef]
32. Wu, C.; Raghavendra, R.; Gupta, U.; Acun, B.; Ardalani, N.; Maeng, K.; Chang, G.; Behram, F.A.; Huang, J.; Bai, C.; et al. Sustainable AI: Environmental Implications, Challenges and Opportunities. *arXiv* **2021**, arXiv:2111.00364. [CrossRef]
33. Patterson, D.; Gonzalez, J.; Le, Q.; Liang, C.; Munguia, L.; Rothchild, D.; So, D.; Texier, M.; Dean, J. Carbon Emissions and Large Neural Network Training. *arXiv* **2021**, arXiv:2104.10350. [CrossRef]
34. EU. Regulation (EU) 2020/852 of the European Parliament and of the Council of 18 June 2020 on the Establishment of a Framework to Facilitate Sustainable Investment, and Amending Regulation (EU) 2019/2088 (Text with EEA relevance). *Off. J. Eur. Communities* **2020**, *198*, 13–43. Available online: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32020R0852&from=EN (accessed on 13 June 2024).
35. EU, AI Act. 2024. Available online: https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai (accessed on 13 June 2024).
36. Darwin, C. On the Origin of Species by Means of Natural Selection. New York: D. Appleton and Company, 443 & 445 Broadway. MDCCCLXI. Available online: https://darwin-online.org.uk/converted/pdf/1861_OriginNY_F382.pdf (accessed on 13 June 2024).
37. Pinto, P.; Bispo, J.; Cardoso, J.M.P.; Barbosa, J.G.; Gadioli, D.; Palermo, G.; Martinovic, J.; Golasowski, M.; Slaninova, K.; Cmar, R.; et al. Pegasus: Performance Engineering for Software Applications Targeting HPC Systems. *IEEE Trans. Softw. Eng.* **2022**, *48*, 732–754. [CrossRef]
38. GG, Sorting Algorithms. 2024. Available online: https://www.geeksforgeeks.org/sorting-algorithms/ (accessed on 13 June 2024).

39. Penev, K. Free Search–comparative analysis 100. *Int. J. Metaheuristics* **2014**, *3*, 118–132. [CrossRef]
40. Jain, S.; Saha, A. Improving and comparing performance of machine learning classifiers optimized by swarm intelligent algorithms for code smell detection. *Sci. Comput. Program.* **2024**, *237*, 103140. [CrossRef]
41. Deng, L. Artificial Intelligence in the Rising Wave of Deep Learning: The Historical Path and Future Outlook [Perspectives]. *IEEE Signal Process. Mag.* **2018**, *35*, 177–180. [CrossRef]
42. Legg, S.; Hutter, M.A. A Collection of Definitions of Intelligence. In *Proceedings of the 2007 Conference on Advances in Artificial General Intelligence: Concepts, Architectures and Algorithms: Proceedings of the AGI Workshop 2006*; IOS Press: Amsterdam, The Netherlands, 2007; pp. 17–24. [CrossRef]
43. Griewank, A.O. Generalized Decent for Global Optimization. *J. Optim. Theory Appl.* **1981**, *34*, 11–39. [CrossRef]
44. Michalewicz, Z. *Genetic Algorithms + Data Structures = Evolution Programs*; Springer: Berlin/Heidelberg, Germany, 1992. [CrossRef]
45. Penev, K. Free Search in Multidimensional Space II. In *Numerical Methods and Applications*; Dimov, I., Fidanova, S., Lirkov, I., Eds.; NMA 2014, Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2015; Volume 8962, pp. 103–111. [CrossRef]
46. Mühlenbein, H.; Schomisch, M.; Born, J. The Parallel Genetic Algorithm as Function Optimizer. *Parallel Comput.* **1991**, *17*, 619–632. [CrossRef]
47. Rosenbrock, H. An automate method for finding the greatest or least value of a function. *Comput. J.* **1960**, *3*, 175–184. [CrossRef]
48. Schwefel, H.P. *Numerical Optimization of Computer Models*; John Wiley & Sons: Chichester, UK, 1981.
49. De Jong, K. An Analysis of the Behaviour of a Class of Genetic Adaptive Systems. Ph.D. Thesis, University of Michigan, Ann Arbor, MI, USA, 1975.
50. Kennedy, J.; Eberhart, R. Particle Swarm Optimisation. In Proceedings of the IEEE International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948. [CrossRef]
51. Storn, R.; Price, K. Differential Evolution—A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
52. Penev, K. Adaptive Heuristic Applied to Large Constraint Optimisation Problem. In *Large-Scale Scientific Computing*; Lirkov, I., Margenov, S., Waśniewski, J., Eds.; LSSC 2007. Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2008; Volume 4818. [CrossRef]
53. GMM-DDS108 (KWE-PM01) Digital Power Consumption Energy Meter UK Plug Socket. 2024. Available online: https://testmeter.sg/webshaper/pcm/files/Data%20Sheet/GMM-DDS108-KWE-PM01-UK.pdf (accessed on 17 June 2024).
54. CUPID, CPU-Z for Windows® x86/x64. 2024. Available online: https://www.cpuid.com/softwares/cpu-z.html (accessed on 17 June 2024).
55. CoreTemp, Core Temp 1.18.1. 2024. Available online: https://www.alcpu.com/CoreTemp/ (accessed on 17 June 2024).
56. Stroll, A.; Martinich, A.P. "Epistemology". Encyclopedia Britannica. 19 April 2024. Available online: https://www.britannica.com/topic/epistemology (accessed on 18 June 2024).
57. Steup, M.; Neta, R. Epistemology. In *The Stanford Encyclopedia of Philosophy (Spring 2024 Edition)*; Zalta, E.N., Nodelman, U., Eds.; Metaphysics Research Lab, Stanford University: Stanford, CA, USA, 2024; Available online: https://plato.stanford.edu/archives/spr2024/entries/epistemology/ (accessed on 24 September 2024).
58. Zins, C. Conceptual approaches for defining data, information, and knowledge. *J. Am. Soc. Inf. Sci. Technol.* **2007**, *58*, 479–493. [CrossRef]
59. Rowley, J. The wisdom hierarchy: Representations of the DIKW hierarchy. *J. Inf. Sci.* **2007**, *33*, 163–180. [CrossRef]
60. Locke, J. *An Essay Concerning Human Understanding*; Hackett Publishing Company: Indianapolis, IN, USA, 1689; ISBN 0-87220-217-8.
61. Nonaka, I.; Takeuchi, H. *The Knowledge—Creating Company: How Japanese Companies Create the Dynamics of Information*; Oxford University Press: Oxford, UK, 1995; ISBN 0-19-509269-4.
62. Davenport, T.H.; Prusak, L. *Working Knowledge: How Organisations Manage What They Know*; Harvard Business School Press: Boston, MA, USA, 2000; ISBN 0-87584-655-6. [CrossRef]
63. Bernardine, M.; Bonansea, B.M. Encyclopedia of Philosophy(Vol. 2. 2nd ed.). 2006. Available online: https://go.gale.com/ps/retrieve.do?tabID=T003&resultListType=RESULT_LIST&searchResultsType=SingleTab&retrievalId=262472d7-abd8-4772-9096-2e2b8b66c631&hitCount=5&searchType=AdvancedSearchForm&currentPosition=1&docId=GALE%7CCX3446800300&docType=Biography&s (accessed on 24 September 2024).
64. Moody, E.A. Encyclopedia of Philosophy(Vol. 9. 2nd ed.). 2006. Available online: https://go.gale.com/ps/retrieve.do?tabID=T003&resultListType=RESULT_LIST&searchResultsType=SingleTab&retrievalId=4054c37d-abec-4cb0-959a-a0df936e4a4a&hitCount=69&searchType=AdvancedSearchForm&currentPosition=1&docId=GALE%7CCX3446802128&docType=Biography& (accessed on 24 September 2024).