Energy and Performance Analysis of Parallel Heterogeneous Genetic Algorithms under Various CPU and GPU DVFS Governors: A Preliminary Study on Predictive Profiling

Amr Abdelhafez
Department of Computing, South East Technological University
Carlow, Ireland
amr.abdelhafez@setu.ie
School of Computer Science, University College Dublin
Dublin, Ireland
Faculty of Computers and Information, Assiut University
Assiut, Egypt

Alexey Lastovetsky
School of Computer Science, University College Dublin
Dublin, Ireland
alexey.lastovetsky@ucd.ie

Abstract

Parallel heterogeneous computing has emerged as a promising approach for addressing computationally intensive problems. Energy efficiency is a critical concern in high-performance computing, particularly when leveraging hybrid architectures such as CPU-GPU systems. This study aims to provide valuable insight into optimizing the trade-off between energy efficiency, performance, and power governors over hybrid architectures.

In this work, we evaluate a Parallel Heterogeneous Genetic Algorithm (HPIGA) by running it under five Dynamic Voltage and Frequency Scaling (DVFS) configurations, exploring different frequency configurations for both CPU and GPU. These configurations investigate various combinations of CPU and GPU operating modes, including "powersave" and "performance". Through these experiments, we analyze the energy consumption and performance characteristics of the parallel algorithm under fixed computational loads. The results reveal interesting insights into CPU-GPU specific DVFS configurations, where setting the CPU and GPUs to high/low frequencies can significantly reduce dynamic energy usage in certain configurations.

These findings contribute to the development of sustainable computing frameworks by addressing the challenges inherent in frequency scaling and heterogeneous computing environments. This study provides a foundation for future research aimed at developing predictive models and advanced scheduling techniques to further optimize energy efficiency in hybrid CPU/GPU architectures.

Keywords

Genetic Algorithm, Parallel Computing, Heterogeneous Computing, Hybrid Server, Energy Efficiency, GPU, OpenMP, OpenACC, DVFS, DVFS Governors, Performance, Powersave

GECCO '25, July 14–18, 2025, Malaga, Spain

ACM ISBN 979-8-4007-1465-8/2025/07
https://doi.org/10.1145/3712256.3726422

ACM Reference Format:
Amr Abdelhafez and Alexey Lastovetsky. 2025. Energy and Performance Analysis of Parallel Heterogeneous Genetic Algorithms under Various CPU and GPU DVFS Governors: A Preliminary Study on Predictive Profiling. In *Genetic and Evolutionary Computation Conference (GECCO '25), July 14–18, 2025, Malaga, Spain.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3712256.3726422

1 Introduction

Genetic Algorithms (GAs), inspired by natural selection, have been widely adopted to solve complex optimization problems [22]. In real-life scenarios, the execution of GAs can result in significant energy consumption. Parallel Genetic Algorithms (PGAs) provide a promising avenue for addressing these challenges by leveraging parallel computing to distribute workloads and reduce execution times [2]. However, most existing parallel genetic algorithm (PGA) implementations prioritize performance optimization, often neglecting the critical aspect of energy consumption. Furthermore, traditional implementations and energy studies are generally confined to homogeneous platforms, such as one multi-core CPU or one GPU, which limits their applicability in modern hybrid systems [4, 10].

Parallel heterogeneous computing has emerged as a brilliant solution for tackling computationally intensive problems found in real-life environments. As hybrid architectures such as CPU-GPU systems become prevalent, energy efficiency has become a pressing concern. The interplay between performance and energy consumption, particularly under diverse operating conditions between different devices, requires a comprehensive investigation to guide sustainable computing practices [5]. Dynamic Voltage and Frequency Scaling (DVFS) is a widely used technique to reduce energy consumption across different platforms [7]. Many processors offer chip-level DVFS, which adjusts the frequency and voltage of the computing cores. However, the intricate energy consumption dynamics of these systems under varying DVFS configurations in hybrid architectures remain unexplored, leaving a critical gap in understanding their operational efficiency. This challenge is compounded when tackling multi-objective optimization problems with conflicting goals, requiring innovative approaches to balance performance and energy efficiency.

In this work, we evaluate the performance and energy efficiency of a basic parallel Genetic Algorithm (GA) called HPIGA [6] under varying DVFS configurations. HPIGA is a heterogeneous parallel islands-model GA implemented using OpenMP and OpenACC directives to enable execution on multicore processors and accelerators, including GPUs. While most researchers typically use the "ondemand" governor by default, without exploring other power governors, our study investigates the energy consumption and performance of five different frequency configurations: "powersave, "ondemand", "performance", "high-CPU", and "high-GPU". We test our experiments by running them on a hybrid system with a multi-core CPU and two GPUs. By experimenting with various combinations of CPU and GPU frequency governors, we aim to analyze their impact on the energy consumption and performance of island-model PGAs under fixed computational loads. Our findings shed light on the trade-offs between energy efficiency, performance, and power mode in hybrid architectures. Additionally, the insights gained will contribute to future efforts in predictive modeling to estimate energy consumption profiles across broader configurations and workloads.

The key contributions of this work are:

- Our work provides a detailed analysis of the impact of five different DVFS configurations on energy consumption and performance of the parallel islands-model GA, which is the most commonly used model of PGAs, filling a gap in the study of heterogeneous architectures.
- We demonstrate how hybrid systems' diverse operating modes can be leveraged to balance energy efficiency and computational performance, offering valuable insights for future energy-aware computing frameworks.

The findings reveal new strategies for optimizing energy consumption without compromising computational performance, advancing the understanding of energy-aware parallel genetic algorithms over hybrid and heterogeneous architectures.

The rest of the paper is organized as follows: Section 2 reviews related works. Section 3 describes the methodology and experimental setup. Section 4 presents the results and analysis of our findings. Section 5 concludes the study and explores potential directions for future research.

2 Related Works

The quest for energy efficiency in high-performance heterogeneous systems has driven extensive research, particularly on the effects of DVFS on parallel algorithms. Prior studies have examined how different algorithm decisions and DVFS settings influence performance and energy consumption behaviors. This section highlights significant advancements in energy-aware optimization strategies for heterogeneous computing systems.

An energy-aware approach for bi-objective optimization is presented in [8] for heterogeneous CPU-GPU architectures, aiming to minimize runtime and energy consumption. The study introduces a cost function to balance these objectives and evaluates a greedy scheduling algorithm under DVFS mechanisms. Experimental results demonstrate significant energy savings without substantial runtime increases through optimized workload distribution. In contrast, our study here focuses on PGA's energy efficiency and performance under varying DVFS settings, emphasizing island-model distribution. An analysis of DVFS techniques for improving the energy efficiency of GPUs is presented in [20]. The authors there provided a survey of DVFS strategies aimed at enhancing GPU energy efficiency. Their analysis underscores the significance of selecting appropriate DVFS schemes tailored to specific workloads to achieve optimal energy savings. Their study lacks an exploration of CPU or heterogeneous hybrid investigations. Another study examined the role of DVFS in reducing dynamic power consumption by adjusting processor clock frequencies is presented in [15]. They highlight the shift from single-core to heterogeneous platforms, integrating multicore CPUs and GPUs, which has led to the need for application-level energy optimization techniques. The study also reviews component-level energy measurement methods, presenting their trade-offs in accuracy and performance. Lastly, it introduces challenges in scaling energy and performance optimization. Readers interested in understanding the influence of DVFS on energy usage, application performance, and system dynamics are encouraged to refer to [16, 18, 24, 27].

In the area of energy efficiency in metaheuristics, several studies have focused on addressing the challenges of energy consumption and resource utilization. This paragraph reviews various studies that have investigated the energy consumption of search algorithms executed on parallel and heterogeneous systems. In [3], the authors conducted a comparative study on the energy consumption of parallel execution of search problems examining several metaheuristic algorithms, including GA. The study demonstrated that parallel implementations significantly improve computational throughput while managing energy consumption. While this research provides a foundational understanding of the energy consumption of parallel metaheuristics, it did not explore GPU profiles or various DVFS settings. Another study that employs heuristics and addresses energy-efficient scheduling in heterogeneous CPU-GPU systems is presented in [23]. A CPU-GPU utilization model is developed to analyze power consumption and scheduling constraints. The authors propose a heuristic greedy strategy (UEJS) and a hybrid Particle Swarm Optimization algorithm (PSO) to optimize energy efficiency and reduce job rejection rates. A study [12] proposes a two-level parallel genetic algorithm, to solve the N-Queens problem using heuristic search techniques. The approach integrates GAs with Simulated Annealing (SA) to enhance solution exploration. The study also highlights opportunities to improve energy efficiency and computational performance by optimizing CUDA kernel parameters, utilizing CPU cores more effectively, and transitioning to a heterogeneous platform with CPUs and GPUs.

An approach employing a dual heterogeneous PGA is presented in [17], focusing on energy efficiency and production performance in the dynamic energy-aware job shop scheduling problems. The method simultaneously executes a cellular GA on GPUs and a classic GA on multi-core CPUs, leveraging two-level parallelization to minimize energy costs, total tardiness, and schedule disruption. Numerical tests demonstrate significant improvements in energy efficiency, solution quality, and execution time. The study highlights the potential of parallel platforms for addressing large-scale, energy-aware scheduling problems efficiently. Recent advancements have explored parallel and heterogeneous computing strategies to enhance performance in computationally intensive tasks. For instance, a parallel Metaheuristic approach presented in [11] introduces a parallel framework for ensemble feature selection using multi-core CPUs and GPUs. By employing GAs, PSO, and grey wolf optimization, the approach improves accuracy and reduces execution time on 21 large datasets. While their focus is on optimizing ensemble feature selection, their study did not investigate the performance and energy efficiency of PGAs running on hybrid systems. A significant contribution to GPU-based parallel computing is presented in [25]. This study develops a parallel GPU-based GA to optimize scheduling in resource-constrained multi-project scenarios. By leveraging GPU acceleration, the algorithm efficiently solves large-scale instances, demonstrating improved performance and solution quality compared to CPU-based implementations. While their work emphasizes the efficiency of GPU-based GAs, they did not investigate the energy consumption profiles over various GPU frequencies. A recent survey on energy-aware scheduling in high-performance computing (HPC) systems, found in [13], explores various techniques and goals for optimizing energy consumption in CPU-GPU systems. This article discusses mechanisms such as DVFS, power capping, and thread throttling to manage power and energy trade-offs efficiently. It also discusses the role of performance-energy configurations in enhancing energy efficiency in modern HPC environments. This work provides a valuable understanding of the energy-related challenges and advancements within CPU-GPU systems. For further insights into energy efficiency and performance trade-offs of PGAs, we direct readers to [14, 19, 21, 26].

In summary, while previous studies have explored various aspects of energy efficiency and performance in hybrid computing environments, they either focused on a single DVFS setting without detailed configurations or studied different DVFS settings on specific system configurations. Furthermore, prior works often utilized heuristics to optimize or select energy settings without directly studying the efficiency of the underlying algorithms themselves. Our study differs by investigating the performance and energy efficiency of PGAs on a heterogeneous system under various DVFS settings, paving the way for the development of a predictive model for energy consumption in hybrid systems.

3 Experimental Setup

To evaluate the energy efficiency and performance of the PGA, we conducted a systematic analysis under five distinct DVFS configurations. The configurations studied are:

(1) **powersave**: All computing devices (CPUs and GPUs) set to "Powersave" mode.[1]
(2) **ondemand**: All computing devices are set to the "Ondemand" mode.
(3) **performance**: All computing devices are set to the "Performance" mode.
(4) **high-CPU**: The CPU is set to "Performance" mode, while the GPUs are set to "Powersave" mode.
(5) **high-GPU**: The CPU is set to "Powersave" mode, while the GPUs are set to "Performance" mode.

By studying these five configurations, we aim to provide a comprehensive analysis of the trade-offs between energy efficiency and computational performance. The first three configurations: "powersave", "ondemand", and "performance" establish baseline behaviors by assessing extreme and balanced energy-performance trade-offs in homogeneous setups, offering key references for comparison with hybrid configurations. In particular, the "high-CPU" and "high-GPU" modes provide critical insights into the behavior of the system under asymmetric power-performance setups, reflecting real-world scenarios where computational resources operate under varying energy constraints. These configurations allow us to explore how heterogeneous DVFS settings influence load balancing, energy efficiency, and execution time, especially when different devices prioritize performance or energy savings.

By evaluating energy consumption and performance across these configurations, we aim to provide a comprehensive understanding of their effects. This approach allows for a more thorough analysis of energy efficiency. Ultimately, our goal is to highlight how varying governors influence the performance of the GA. This exploration will help understand how different governors affect the energy consumption and performance of the PGA. To focus on studying the effect of DVFS, we divide the islands equally across all devices. If the total number of islands is not divided evenly, we adjust the distribution by adding extra islands to some devices based on the remainder. For consistency and a fair comparison, the stopping condition is defined by the number of function evaluations, which remains the same for each dimension, regardless of the number of islands employed.

3.1 Parameter Settings and System Specifications

We provide an overview of the benchmark problem used to assess the experimental setup and the parameter configurations applied during our runs. The primary objective of this study is to investigate the performance and energy consumption profiles of the PGA under different power governors. To achieve this, we use the well-known One-Max problem, a widely recognized test case for evolutionary algorithms, known for its simple search space and consistent energy and performance characteristics. This simple function should not significantly affect the energy results, ensuring that the energy profiles reflect the performance of the governors rather than the complexity of the problem itself [1].

To ensure reliable results, we vary the dimensions, including lower dimensions (100 and 500) and higher ones (1000, 3000, and 5000), which increase runtime and provide robust energy consumption profiles. Our analysis employs a comprehensive statistical

[1]For the GPUs, the "powersave" and "performance" modes are set by adjusting the clock speeds, streaming multiprocessors (SM), and power consumption (between 100W and 300W) to their minimum or maximum values, respectively.

approach to generate reliable findings for each configuration. Table 1 summarizes the parameter settings used in the experiments, selected based on commonly used values reported in the literature and previous studies. These parameters were determined through a series of preliminary numerical experiments designed to highlight differences between the algorithms being tested.

Table 1: Parameter Settings

Definitions	Values
Sub-population size	50 individual
Recombination	Uniform, $pc = 0.6$
Mutation	Bit-flip, $pm = 0.0001$
Selection	Binary tournament
Replacement	Replacing the worst
Elitism	Yes
Migration interval	Every 10 iterations

We conduct our experiments on a hybrid heterogeneous server platform, detailed in Table 2, which contains a single-socket Intel Icelake multicore CPU and two Nvidia A40 GPUs. The two GPUs are connected to the motherboard and accessible to all cores (0-63) of the CPU. Energy consumption is measured using a WattsUp Pro power meter, which monitors the node's total power usage through the wall socket. Data is collected via a USB interface using non-intrusive Perl scripts. The power meter, with a sampling rate of one sample per second and a minimum measurable power of 0.5 W, is periodically calibrated using a Yokogawa WT210 meter. The measurements collected through physical methods using external power meters are considered highly reliable [9]. To ensure result reliability, experiments are repeated until the response variables (execution time and energy) achieve a 95% confidence interval with 5% precision. Statistical tests, including Student's t-test and Pearson's chi-squared test, validate the independence and normality of observations.

Table 2: System Specifications.

Intel Platinum 8362 Icelake	
No. of cores per socket	32
No. of threads per core	2
Socket(s)	2
L1d cache, L1i cache	1.5 MiB, 1 MiB
L2 cache, L3 cache	40 MiB, 48 MiB
Total main memory	62 GB DDR4-3200
TDP	265 W
NVIDIA A40 GPU	
No. of GPUs	2
No. of Ampere cores	10,752
Total board memory	48 GB GDDR6 (with ECC)
Memory bandwidth	696 GB/sec
TDP	300 W

4 Experimental Results and Analysis

This section presents the experimental results obtained from evaluating the performance and energy efficiency of HPIGA. We present the total energy, dynamic energy, and execution time measured for the dimensions under study. Table 3 presents the total energy consumption values observed under the five DVFS configurations studied. Total energy is a more comprehensive metric for evaluating overall efficiency, as it accounts for the combined impact of dynamic and static energy.

The results obtained from Table 3 reveal significant trade-offs between total energy consumption and the DVFS configurations used. For the dimensions under study, the "ondemand" and "high-CPU" configurations exhibit the lowest total energy consumption in the majority of the cases among the other configurations. The explanation for this behavior lies in the nature of the governors. The "ondemand" governor, being the default system-optimized option, dynamically adjusts the frequency range based on workload demands, balancing performance and energy efficiency effectively. The "high-CPU" governor imposes strict controls on the energy and power usage of GPUs, prioritizing energy savings over performance flexibility. This highlights how the characteristics and configurations of each governor can significantly impact energy consumption, with "ondemand" excelling in adaptive efficiency and "high-CPU" in stringent energy management. Conversely, the "performance" and "powersave" configurations often achieve lower energy consumption for moderate and larger dimensions, aligning with their optimized utilization of processing power to reduce execution time and energy overhead.

The "high-GPU" configuration tends to exhibit the highest total energy consumption (underlined) across most dimensions, indicating its inefficiency for computationally intensive workloads. This behavior can be attributed to the governor setting the GPUs to operate at their highest power and clock levels, resulting in significantly higher static energy consumption while minimizing CPU usage. This imbalance between GPU and CPU energy utilization leads to an inefficient overall energy profile. Additionally, the governor's lack of dynamic adaptability to workload demands further exacerbates its energy inefficiency, as it fails to optimize energy usage across varying computational intensities. To provide deeper insights into the energy patterns, we present the dynamic energy values in Table 4. Dynamic energy refers to the energy consumed during the execution of computational tasks, varying with the system's workload and power states. It is calculated using the formula: Dynamic Energy = Total Energy − (Base Power × Execution Time). Unlike dynamic energy, static energy corresponds to the energy consumed by the system in its idle state, determined by the configuration of the system's infrastructure. Static energy represents a significant portion of the total energy, as it is independent of the workload but depends on the system's design and base power usage. This allows for a better understanding of how dynamic energy contributes to the overall efficiency, highlighting the impact of varying DVFS configurations on energy consumption.

Table 3: Mean of total energy in Joules.

# of Islands	DVFS	Dimensions under the study 100	500	1000	3000	5000
128	powersave	<u>1225</u>	4435	22302	93343	<u>189585</u>
	ondemand	425	2404	**7573**	55921	**133356**
	performance	**393**	2341	17177	62723	146965
	high-CPU	471	**2205**	14724	**50893**	134097
	high-GPU	1184	<u>5174</u>	<u>29569</u>	<u>98832</u>	189023
64	powersave	2662	4438	11894	78868	162867
	ondemand	**557**	2020	**5945**	**50194**	**124135**
	performance	648	**1819**	6902	59960	156913
	high-CPU	749	1981	6685	56481	136828
	high-GPU	<u>3983</u>	<u>5259</u>	<u>14066</u>	<u>111365</u>	<u>206162</u>
32	powersave	3836	4432	6321	72391	173376
	ondemand	2821	2069	6006	**48819**	**81515**
	performance	2176	2082	2733	61110	146031
	high-CPU	**224**	**1870**	**2649**	49855	132435
	high-GPU	<u>5207</u>	<u>5599</u>	<u>7122</u>	<u>94746</u>	<u>255328</u>
16	powersave	5242	22507	3590	83017	184520
	ondemand	**1094**	2957	<u>4940</u>	47498	**105897**
	performance	1472	3210	2208	56911	141413
	high-CPU	1579	**2524**	**2176**	**44611**	128891
	high-GPU	<u>6946</u>	<u>27168</u>	3475	<u>93295</u>	<u>235028</u>

Boldfaced and underlined represent the least and highest values per dimension, respectively

Table 4: Mean of dynamic energy in Joules.

# of Islands	DVFS	Dimensions under the study 100	500	1000	3000	5000
128	powersave	<u>372</u>	497	2731	11706	23878
	ondemand	195	<u>1084</u>	4057	<u>29757</u>	<u>71266</u>
	performance	**121**	445	<u>5655</u>	23187	48577
	high-CPU	221	702	5582	17940	53713
	high-GPU	172	**150**	**2324**	**8095**	**16250**
64	powersave	435	532	1430	9867	20320
	ondemand	186	<u>1013</u>	<u>2942</u>	<u>25532</u>	<u>64607</u>
	performance	**177**	**190**	1870	19663	51807
	high-CPU	378	695	2607	22546	54714
	high-GPU	<u>495</u>	188	**924**	**9279**	**9605**
32	powersave	665	501	848	8970	21541
	ondemand	<u>1797</u>	<u>1053</u>	<u>3255</u>	<u>25902</u>	43167
	performance	591	436	478	24394	48469
	high-CPU	950	544	996	19952	<u>53748</u>
	high-GPU	**481**	**124**	**445**	**7892**	**21461**
16	powersave	<u>2545</u>	<u>18213</u>	892	10259	22959
	ondemand	**484**	1782	<u>4209</u>	<u>24126</u>	<u>56233</u>
	performance	617	1322	1076	22135	47051
	high-CPU	811	**857**	1231	15086	54075
	high-GPU	765	2903	**57**	**7722**	**19789**

Boldfaced and underlined represent the least and highest values per dimension, respectively

Table 4 shows that across all dimensions, the choice of DVFS configuration significantly influences dynamic energy consumption. The "high-GPU" consistently achieves the lowest energy consumption in most configurations, particularly for larger dimensions. This behavior arises because GPUs are optimized for parallel processing and energy-efficient operations, whereas CPUs often consume more power when handling similar workloads. Conversely, "ondemand" and "powersave" exhibit the highest energy consumption in several scenarios, indicating their inefficiency for this workload. Notably, the "performance" configuration often achieves a balance, offering competitive energy efficiency across different dimensions. These findings emphasize the impact of DVFS settings on energy usage, guiding optimal governor selection based on computational demands. "High-CPU" configuration shows moderate energy consumption, reflecting its reliance on CPU-intensive operations that are less energy-efficient than GPU-based processing. These observations provide a foundation for analyzing the impact of DVFS settings on execution time, as detailed in Table 5.

The execution time results reveal the performance characteristics of different DVFS configurations across varying dimensions and island configurations. The "performance" configuration consistently achieves the shortest execution times, showcasing its efficiency in high-demand computational tasks due to its operation at maximum clock frequencies, which minimizes computation delays. In contrast, "powersave" exhibits the highest execution times in most scenarios, reflecting its trade-off of energy efficiency at the cost of performance. This behavior arises because the "powersave" governor locks the CPU at its minimum frequency, leading to slower task completion as the computational power is significantly reduced to conserve energy. The "ondemand" configuration provides a balanced approach, delivering competitive times in several configurations but falling short of performance's optimal efficiency. Interestingly, the "high-CPU" configuration closely aligns with the "performance" configuration, highlighting the CPU's ability to handle intensive workloads efficiently. The "high-GPU" configuration optimized for energy efficiency shows longer execution times, especially for smaller dimensions, indicating possible inefficiencies when processing lighter workloads. To visually present our results, we provide Figure 1, which demonstrates the energy consumption across the different DVFS and island configurations.

Figure 1 illustrates the dynamic and total energy consumption for different dimensions under various DVFS configurations. A clear trend emerges, showing that smaller dimensions generally lead to higher dynamic energy consumption across most DVFS configurations as a result of increased CPU utilization. Conversely, larger dimensions tend to benefit from GPU optimization, resulting in lower dynamic energy usage, particularly under the "high-GPU" configuration. However, the "high-GPU" configuration also demonstrates the highest total energy consumption because of its high static energy overhead. On the other hand, "ondemand" and "high-CPU" configurations maintain balanced energy profiles across dimensions, with lower total energy consumption, highlighting their adaptive and energy-efficient behavior. These patterns emphasize the importance of the size of the workload in determining the energy efficiency of DVFS settings.

Overall, we can observe that there is no single optimal configuration of governors for energy and performance under the fixed, equally distributed workload. For example, in dimension 5000, the "ondemand" configuration achieved the lowest total energy consumption across all tested numbers of islands but exhibited the highest dynamic energy consumption. However, in dimension 500, "ondemand" configuration performed poorly in terms of total energy. Additionally, the "high-GPU" configuration demonstrated the lowest dynamic energy consumption in most cases (15 out of 20) but failed to achieve the best execution times in any scenario. To address this, workloads should be optimally distributed according to the frequency-state updates to achieve better load balancing. These observations highlight the necessity for bi-objective optimization to identify Pareto-optimal fronts that account for both energy efficiency and performance while considering an optimal workload distribution.

5 Conclusions and Future Work

In this study, we evaluated the energy efficiency and performance of a Parallel Heterogeneous Genetic Algorithm (HPIGA) under five different DVFS configurations, specifically focusing on "ondemand", "power-save", "performance", "high-CPU", and "high-GPU" modes for CPU and GPU frequency configurations. Through a series of experiments, we analyzed the energy consumption and performance trade-offs across varying frequencies, highlighting the potential for optimizing energy efficiency while maintaining computational performance.

The results show that the "ondemand" configuration demonstrates a commendable balance by dynamically adjusting frequencies to align with workload demands, thereby optimizing energy efficiency without significant performance degradation. However, this level of optimization may unintentionally worsen execution times, as it attempts to balance performance dynamically. This unexpected side effect is one of the first to be reported in heterogeneous systems. Similarly, the "high-CPU" configuration achieves low total energy consumption by imposing stringent controls on energy and power usage, prioritizing energy savings. However, no single configuration proves to be an optimal solution for the bi-objective optimization of energy and time. For instance, the "high-GPU" configuration, despite its higher total energy consumption and execution time, achieves lower dynamic energy consumption. This makes it a suitable option for long-running tasks on servers with low utilization, such as during periods of inactivity or off-peak hours. These findings underscore the necessity of selecting appropriate DVFS configurations tailored to specific workload characteristics to achieve optimal energy-performance trade-offs.

Modern systems contain CPUs with several different governors and varying options for GPUs based on the driver and hardware. These governors combine to offer a wide range of frequency configuration possibilities. In future work, we aim to expand this research

Table 5: Mean execution time in seconds.

		Dimensions under the study				
# of Islands	**DVFS**	**100**	**500**	**1000**	**3000**	**5000**
128	powersave	2.97	13.72	68.19	284.45	577.38
	ondemand	0.91	5.22	**13.90**	103.41	245.41
	performance	**0.63**	**4.39**	26.67	**91.52**	**227.75**
	high-CPU	0.75	4.50	27.37	98.66	240.67
	high-GPU	2.61	12.95	70.22	233.86	445.29
64	powersave	7.76	13.61	36.46	240.42	496.68
	ondemand	1.47	3.98	11.87	97.48	**235.29**
	performance	**1.09**	**3.77**	**11.65**	**93.28**	243.30
	high-CPU	1.11	3.85	12.21	101.60	245.85
	high-GPU	8.99	13.07	33.87	263.11	506.59
32	powersave	11.05	13.70	19.07	220.98	529.04
	ondemand	4.05	4.02	10.87	90.58	**151.57**
	performance	**3.67**	**3.81**	5.22	**84.99**	225.84
	high-CPU	3.88	3.97	**4.95**	89.53	235.59
	high-GPU	12.18	14.11	17.21	223.85	602.75
16	powersave	9.40	14.96	9.40	253.51	562.93
	ondemand	2.41	4.64	2.89	92.38	**196.30**
	performance	**1.98**	**4.37**	**2.62**	**80.50**	218.43
	high-CPU	2.30	4.99	2.83	88.40	224.00
	high-GPU	15.93	62.54	8.81	220.55	554.74

Boldfaced and underlined represent the least and highest values per dimension, respectively

by exploring various frequency configurations and developing predictive models that go beyond using work distribution as a sole decision variable. These models will identify Pareto-optimal solutions for energy efficiency and performance, providing a robust framework for energy-efficient computing in heterogeneous systems. This approach will enable more accurate energy optimization strategies, supporting the development of highly efficient parallel algorithms for diverse workloads. Furthermore, we are expanding our research to include the configuration of the algorithm, the platform setup, and the application itself, enabling it to identify the optimal configurations. Moreover, we aim to investigate the impact of dynamically distributing the workload on the efficiency of the default system governor.

Acknowledgments

This publication has emanated from research conducted with the financial support of TaighdeÉireann – Research Ireland under Grant number 20/FFP-P/8683.

References

[1] Amr Abdelhafez and Enrique Alba. 2017. Speed-up of synchronous and asynchronous distributed Genetic Algorithms: A first common approach on multiprocessors. *2017 IEEE Congress on Evolutionary Computation (CEC)* (2017). https://doi.org/10.1109/CEC.2017.7969632

[2] Amr Abdelhafez, Enrique Alba, and Gabriel Luque. 2019. Performance analysis of synchronous and asynchronous distributed genetic algorithms on multiprocessors. *Swarm and Evolutionary Computation* 49 (2019), 147–157. https://doi.org/10.1016/j.swevo.2019.06.003

[3] Amr Abdelhafez, Enrique Alba, and Gabriel Luque. 2020. Parallel execution combinatorics with metaheuristics: Comparative study. *Swarm and Evolutionary Computation* 55 (2020), 100692. https://doi.org/10.1016/j.swevo.2020.100692

[4] Amr Abdelhafez, Gabriel Luque, and Enrique Alba. 2019. Analyzing the energy consumption of sequential and parallel metaheuristics. *2019 International Conference on High Performance Computing & Simulation (HPCS)* (2019). https://doi.org/10.1109/hpcs48598.2019.9188170

[5] Amr Abdelhafez, Gabriel Luque, and Enrique Alba. 2019. A component-based study of energy consumption for sequential and parallel genetic algorithms. *The Journal of Supercomputing* 75, 10 (2019), 6194–6219. https://doi.org/10.1007/s11227-019-02843-4

[6] Amr Abdelhafez, Ravi Reddy Manumachu, and Alexey Lastovetsky. 2024. Parallel genetic algorithms on hybrid servers: Design, implementation, and optimization for performance and Energy. Preprint Available at SSRN. (2024). https://doi.org/10.2139/ssrn.5056740

[7] Bilge Acun, Kavitha Chandrasekar, and Laxmikant V Kale. 2019. Fine-grained energy efficiency using per-core dvfs with an adaptive runtime system. In *2019 Tenth International Green and Sustainable Computing Conference (IGSC)*. IEEE, 1–8.

[8] Juan José Escobar, Julio Ortega, Antonio Francisco Díaz, Jesús González, and Miguel Damas. 2019. Energy-aware load balancing of parallel evolutionary algorithms with heavy fitness functions in heterogeneous CPU-GPU architectures. *Concurrency and Computation: Practice and Experience* 31, 6 (2019), e4688.

[9] Muhammad Fahad, Arsalan Shahid, Ravi Reddy, and Alexey Lastovetsky. 2019. A Comparative Study of Methods for Measurement of Energy of Computing. *Energies* 12, 11 (2019). https://doi.org/10.3390/en12112204

[10] Tomohiro Harada and Enrique Alba. 2020. Parallel Genetic Algorithms: A Useful Survey. 53, 4, Article 86 (2020), 39 pages. https://doi.org/10.1145/3400031

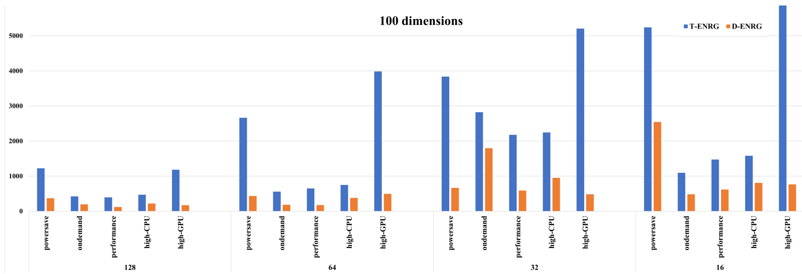
[11] Neveen Mohammed Hijazi, Hossam Faris, and Ibrahim Aljarah. 2021. A parallel metaheuristic approach for ensemble feature selection based on multi-core architectures. *Expert Systems with Applications* 182 (2021), 115290. https://doi.org/10.1016/j.eswa.2021.115290

[12] Cao Jianli, Chen Zhikui, Wang Yuxin, and Guo He. 2020. Parallel genetic algorithm for N-Queens problem based on message passing interface-compute unified device architecture. *Computational Intelligence* 36, 4 (2020), 1621–1637.

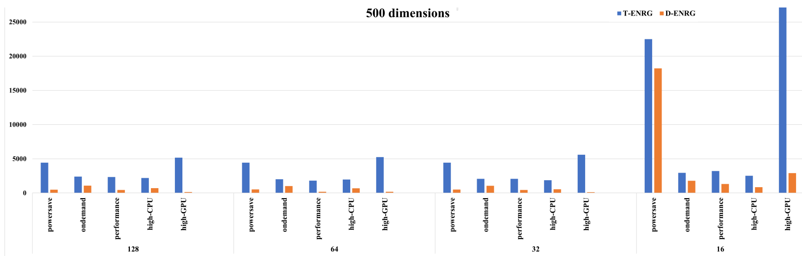
[13] Bartłomiej Kocot, Paweł Czarnul, and Jerzy Proficz. 2023. Energy-aware scheduling for high-performance computing systems: A survey. *Energies* 16, 2 (2023), 890.

[14] Neetesh Kumar and Deo Prakash Vidyarthi. 2017. A GA based energy aware scheduler for DVFS enabled multicore systems. *Computing* 99 (2017), 955–977.

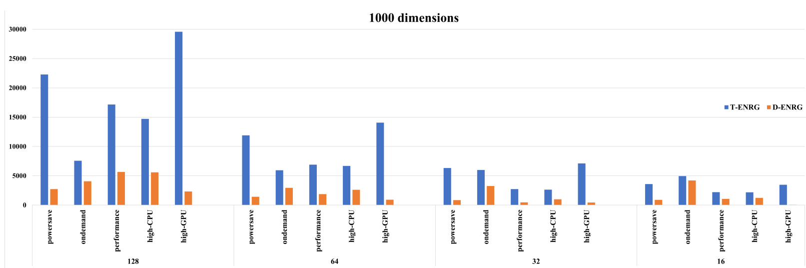
[15] Alexey Lastovetsky and Ravi Reddy Manumachu. 2023. Energy-efficient parallel computing: Challenges to scaling. *Information* 14, 4 (2023), 248.



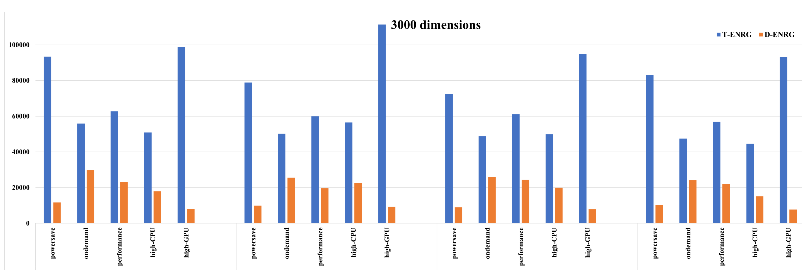
(a) Energy consumption for 100 Dimensions



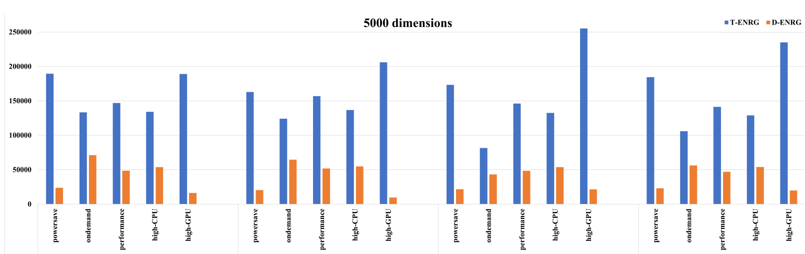
(b) Energy consumption for 500 Dimensions



(c) Energy consumption for 1000 Dimensions



(d) Energy consumption for 3000 Dimensions



(e) Energy consumption for 5000 Dimensions

Figure 1: Total and dynamic energy consumption for the dimensions under study with different DVFS configurations.

[16] Zengpeng Li, Huiqun Yu, Guisheng Fan, Jiayin Zhang, and Jin Xu. 2024. Energy-efficient offloading for DNN-based applications in edge-cloud computing: A hybrid chaotic evolutionary approach. *J. Parallel and Distrib. Comput.* 187 (2024), 104850.

[17] Jia Luo, Didier El Baz, Rui Xue, and Jinglu Hu. 2020. Solving the dynamic energy aware job shop scheduling problem with the heterogeneous parallel genetic algorithm. *Future Generation Computer Systems* 108 (2020), 119–134.

[18] Kai Ma, Xue Li, Wei Chen, Chi Zhang, and Xiaorui Wang. 2012. Greengpu: A holistic approach to energy efficiency in gpu-cpu heterogeneous architectures. In *2012 41st international conference on parallel processing*. IEEE, 48–57.

[19] Amjad Mahmood, Salman A Khan, Fawzi Albalooshi, and Noor Awwad. 2017. Energy-aware real-time task scheduling in multiprocessor systems using a hybrid genetic algorithm. *Electronics* 6, 2 (2017), 40.

[20] Ashish Mishra and Nilay Khare. 2015. Analysis of DVFS techniques for improving the GPU energy efficiency. *Open Journal of Energy Efficiency* 4, 4 (2015), 77–86.

[21] Anju S Pillai, Kaumudi Singh, Vijayalakshmi Saravanan, Alagan Anpalagan, Isaac Woungang, and Leonard Barolli. 2018. A genetic algorithm-based method for optimizing the energy consumption and performance of multiprocessor systems. *Soft Computing* 22 (2018), 3271–3285.

[22] El-Ghazali Talbi. 2009. *Metaheuristics from design to implementation.* John Wiley & Sons.

[23] Xiaoyong Tang and Zhuojun Fu. 2020. CPU–GPU utilization aware energy-efficient scheduling algorithm on heterogeneous computing systems. *IEEE Access* 8 (2020), 58948–58958.

[24] Zhuo Tang, Ling Qi, Zhenzhen Cheng, Kenli Li, Samee U Khan, and Keqin Li. 2016. An energy-efficient task scheduling algorithm in DVFS-enabled cloud environment. *Journal of Grid Computing* 14 (2016), 55–74.

[25] Furkan Uysal, Rifat Sonmez, and Selcuk Kursat Isleyen. 2021. A graphical processing unit-based parallel hybrid genetic algorithm for resource-constrained multi-project scheduling problem. *Concurrency and Computation: Practice and Experience* 33, 16 (2021), e6266.

[26] Yonghee Yun, Eun Ju Hwang, and Young Hwan Kim. 2019. Adaptive genetic algorithm for energy-efficient task scheduling on asymmetric multiprocessor system-on-chip. *Microprocessors and Microsystems* 66 (2019), 19–30.

[27] Josip Zidar, Tomislav Matić, Ivan Aleksi, and Željko Hocenski. 2024. Dynamic Voltage and Frequency Scaling as a Method for Reducing Energy Consumption in Ultra-Low-Power Embedded Systems. *Electronics* 13, 5 (2024), 826.