

Introducing the Hundredth Monkey Effect: An Efficient Metaheuristic for Fast Convergence in the Least Squares Minimization Problem

Yashar Salami, IEEE Professional Member

Faculty of Computer and Information Technologies, Cappadocia University, Ürgüp, Nevşehir, Turkey
yashar.salami@kapadokya.edu.tr

Abstract—In the face of escalating demand for high-speed optimization algorithms, particularly for complex and time-sensitive problems, developing a rapid metaheuristic method is a significant challenge. This paper introduces the Hundredth Monkey Effect as a new metaheuristic approach focusing on enhancing speed and convergence efficiency in solving optimization problems, specifically the Least Squares Minimization Problem. Simulations show that the Hundredth Monkey Effect algorithm outperforms Genetic Algorithms (GA), Simulated Annealing (SA), and Bee Colony Optimization (BCO) in terms of rapid convergence. This substantial improvement in both speed and convergence underscores the practicality and efficiency of the Hundredth Monkey Effect in addressing complex, time-sensitive optimization problems.

Keywords— *Hundredth Monkey Effect, Optimization, Meta-Heuristic, Algorithms, Machine learning.*

I. INTRODUCTION

In today's rapidly evolving world, where technologies such as the Internet of Things (IoT), the Internet of Vehicles (IoV), and blockchain are reshaping the digital landscape, computational complexity and optimization challenges are continuously increasing [1][2]. Among these technologies, blockchain plays a crucial role in ensuring security, transparency, and trust across decentralized IoT and IoV networks, further intensifying the demand for efficient optimization methods [3][4]. The quest for efficient, optimal solutions has become a fundamental necessity across domains such as engineering, computer science, mathematics, and economics to ensure secure communication, intelligent decision-making, and trustworthy data management across interconnected systems [5][6]. These problems often involve vast search spaces and many possible solutions, making their optimal solution through precise analytical methods impractical due to time and computational resource constraints [7]. In this context, the emergence of metaheuristic algorithms is crucial, providing highly effective tools to address such challenges and keep us informed and up-to-date with the latest developments [8].

Metaheuristic algorithms, unlike traditional exact methods, are inspired by natural phenomena and collective behaviors of living organisms[9][10]. These algorithms draw from natural patterns, such as Darwinian evolution, collective bird or fish migration, the foraging behavior of bees, and even physical processes like SA[11]. For instance, the GA mimics biological evolution principles such as natural selection and mutation to improve a population of candidate solutions

iteratively, moving closer to an optimal solution[12]. Similarly, Particle Swarm Optimization (PSO), inspired by the flocking behavior of birds, attempts to find optimal solutions through collaboration and interaction among multiple particles, mimicking group dynamics in nature[13]. Their adaptability to various problem domains and practicality in real-world applications reassure us of their effectiveness in solving complex problems, instilling confidence in their use [14].

One of the key advantages of metaheuristic algorithms is their ability to provide acceptable solutions without the need for an exhaustive search of the entire solution space[15]. By employing local and global search techniques and leveraging information gathered from multiple iterations, these algorithms progressively explore the problem space and converge toward optimal or near-optimal solutions[16]. In the case of NP (Nondeterministic Polynomial time) problems, where finding an exact solution can be highly time-consuming, these algorithms are an efficient alternative to traditional methods due to their speed and adaptability[17].

Applying meta-heuristic algorithms spans various industries and domains, from optimizing complex network designs to enhancing system performance. These algorithms have become indispensable tools, inspiring organizations and researchers to achieve their objectives more efficiently[18].

However, one of the significant challenges in employing these algorithms is the time-consuming nature of their search process[19]. Due to the computational complexity and the large volume of data involved, executing these algorithms can be time-intensive, particularly in real-time applications or when computational resources are limited. Thus, developing high-speed metaheuristic algorithms that can be effectively applied in time-sensitive scenarios has become crucial[20].

Exploring high-speed metaheuristic methods addresses the issue of computational delay and opens up new opportunities for applying these algorithms in real-time optimization problems[21]. Therefore, the need for innovation and advancement in the design of fast and efficient metaheuristic algorithms remains one of the most pressing challenges in optimization[22].

A. Contribution

The Hundredth Monkey Effect can inspire the design of a new metaheuristic algorithm where optimal behavior gradually spreads from a small group to the entire population. This effect illustrates a process in which a behavior or

knowledge, once reaching a "critical point," rapidly disseminates in the whole population. Drawing from this concept, we have developed the Hundredth Monkey Effect algorithm, which was then applied to the Least Squares Minimization problem and compared with similar algorithms. By leveraging the rapid spread of behavior or knowledge once a critical threshold is reached, this algorithm can find solutions more quickly.

B. Paper organization

The remainder of the paper is structured as follows: Section Two outlines the problem statement, Section Three introduces the Hundredth Monkey Effect, and Section Four presents the simulation results. The paper concludes with a final summary and conclusions.

II. PROBLEM STATEMENT

A. Least Squares Minimization

The Least Squares Minimization Problem is a fundamental mathematical approach widely used across fields such as statistics, machine learning, and engineering to approximate solutions to overdetermined systems.[23] In essence, it is a method for minimizing the sum of the squares of the residuals, the differences between observed and calculated values of a model[24]. The goal is to find the optimal solution by reducing the error between the predicted outcomes and the actual data points. Mathematically, the least squares problem can be defined as follows:

Given a set of observations $(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)$ and a model $y = f(x; \theta)$, where θ represents the parameters to be estimated, the least squares solution seeks to minimize the objective function Eq (1).

$$\theta \text{ mini} = \sum_{i=0}^n (y_i - f(x_i; \theta))^2 \quad (1)$$

In matrix form, if we consider a system $Ax = b$, where A is an $m \times n$ times matrix, x is a vector of unknowns and b is a vector of observations, the least squares solution is the vector x that minimizes the Euclidean norm of $r = Ax - b$. This can be expressed as Eq 2.

$$\min_x \| Ax - b \|^2 \quad (2)$$

B. Challenges in Least Squares Minimization

Despite its wide applicability, the least squares minimization problem often faces several challenges, particularly when dealing with large datasets, non-linear models, or ill-conditioned systems where small changes in the data can lead to significant variations in the solution. These issues often necessitate the development of robust algorithms that can efficiently converge to an optimal solution[25].

Classical methods such as Gaussian elimination, QR decomposition, and SVD (Singular Value Decomposition) are commonly used to solve linear least-squares problems. However, these methods can become computationally

expensive and prone to local minima when dealing with complex, non-linear, or high-dimensional problems.

C. Need for Meta-Heuristic Approaches

Given the limitations of traditional optimization techniques for handling non-convex, non-linear, or large-scale least-squares minimization problems, meta-heuristic algorithms have emerged as powerful alternatives. Meta-heuristics, such as GA, particle swarm optimization (PSO), and ant colony optimization (ACO), offer more flexible, adaptive search strategies to explore the solution space effectively[26].

By utilizing global search techniques, meta-heuristic methods can overcome the pitfalls of local minima and are particularly suited for high-dimensional problems where classical methods struggle. This lays the foundation for novel optimization frameworks, such as the Hundredth Monkey Effect Algorithm, to address these challenges in the least-squares minimization domain.

III. HUNDREDTH MONKEY EFFECT

The Hundredth Monkey Effect refers to an observation of social behavior among Japanese macaques on the island of Koshima in the 1950s. In these studies, one monkey developed a novel method of washing sweet potatoes in water before consuming them. Only a few other monkeys initially imitated this behavior, but gradually, it spread among the broader population of monkeys.

This observation is often described in such a way that after a certain number of monkeys learned the behavior (commonly referred to as the "hundredth monkey"), the behavior suddenly spread rapidly among the remaining monkeys, even on neighboring islands where there was no direct contact with the original group. This theory explains how a behavior or idea can spread rapidly after reaching a "critical mass" within a community.

The theory has faced criticism due to the need for strong scientific evidence. Many scientists believe this phenomenon can be easily explained by natural social learning and does not require the hypothesis of "collective consciousness" or some form of abnormal communication. Additionally, claims about the spread of the behavior to other islands have not been substantiated.

Despite being scientifically questioned, the theory has gained popularity in public culture and fields such as psychology and sociology, symbolizing how sudden changes occur after reaching a critical threshold or undergoing a significant transformation. The Hundredth Monkey Effect is considered more of a metaphor than a scientific fact, illustrating that social changes can spread rapidly once a certain threshold is reached.

A. The Meta-Heuristic Algorithm Inspired by the Hundredth Monkey Effect

The Hundredth Monkey Effect can inspire the design of a novel meta-heuristic algorithm in which optimal behavior gradually spreads from a small group to the entire population.

The Hundredth Monkey Effect demonstrates a process in which a behavior or knowledge spreads rapidly among the

entire population after reaching a "critical point." This concept can be applied to model collective learning and evolution in meta-heuristic algorithms.

B. General Structure of the Proposed Algorithm:

1. **Initial Population:** The initial population consists of random solutions representing an individual or agent.
2. **Knowledge Dissemination:** In the early stages, each agent independently attempts to improve based on the objective function. Only a limited number of agents initially achieve optimal behavior, known as the "early monkeys."
3. **Critical Point:** Once a certain number of agents (for instance, the hundredth agent) reach a desirable level of optimality, the algorithm enters a new phase. In this stage, optimal knowledge is shared collectively among all agents, moving beyond local optimization.
4. **Collective Behavior Transmission:** After reaching the critical point, the optimal information or behavior spreads across the entire population, allowing other agents to utilize this knowledge to improve their search processes.
5. **Global Optimization:** The population moves toward convergence to an optimal solution following the collective knowledge transfer. However, some agents continue exploring other areas of the search space to avoid getting trapped in local optima.

C. Algorithm Steps

1. **Population Initialization:** Generate a set of random initial solutions.
2. **Solution Evaluation:** Each solution is evaluated and scored based on the objective function.
3. **Gradual Improvement:** Solutions are improved through various methods, such as mutation, crossover, or refinement.
4. **Critical Point:** Knowledge is transferred collectively after achieving a certain number of optimal solutions.
5. **Utilization of Collective Knowledge:** The population generates further improvements using the shared knowledge.
6. **Termination:** The algorithm terminates after meeting specific criteria, such as a fixed number of iterations or the absence of further improvements.

Figure 1 shows the pseudo-code for the Hundredth Monkey Effect.

The Hundredth Monkey Effect Meta-Heuristic Algorithm

step1. Initialize parameters:

- population_size: Number of individuals in the population
- max_iterations: Maximum number of iterations
- knowledge_threshold: Threshold to trigger knowledge sharing
- threshold_value: Value at which optimization is considered complete

Step2. Generate the initial population randomly.

Step3. Evaluate the objective function for each individual in the population.

Step4. Store the best solution (elite solution) from the initial population.

Step5. Begin iteration loop (for iteration = 1 to max_iterations):

- 5.1** For each individual in the population:
 - Generate a new solution by adding random noise.
 - Evaluate the objective function for the new solution.
 - If the new solution is better, replace the current solution with the new one.
- 5.2** Store the best fitness value at this iteration.
- 5.3** Update the elite_solution based on the current population.
- 5.4** For each individual in the population:
 - Move each individual closer to the elite_solution by updating it as:
 $population(i) = 0.9 * population(i) + 0.1 * elite_solution.$
- 5.5** If the number of individuals with fitness below the threshold_value exceeds the knowledge_threshold and knowledge has not been shared:
 - Share knowledge across the population by updating as:
 $population(i) = 0.8 * population(i) + 0.2 * elite_solution.$
 - Mark knowledge_shared as true.

Step6. End iteration loop.

Step7. Output the final elite_solution as the best-found solution.

Fig1.Pseudo-code Hundredth Monkey Effect.

IV. SIMULATION

The simulations were conducted on a Dell E6430 laptop featuring an Intel Core i7 processor, a high-performance component capable of easily handling complex computational tasks. The system's 8GB of RAM further contributed to efficient multitasking and the smooth execution of large-scale applications.

MATLAB 2013 was utilized for the simulations due to its comprehensive suite of tools and libraries for numerical analysis, matrix computations, data processing, and system simulation. MATLAB's versatility made it an optimal choice for executing the research tasks effectively.

A. Parameter settings

Table 1 presents the parameter settings of the optimization algorithms used in this study. These parameters were uniformly applied across all algorithms to ensure accurate and fair simulations and comparisons. The selected settings were used to evaluate and analyze the performance of the proposed method compared to other existing methods.

Table 1 Parameter Settings.

Algorithm	Parameter	Value
Hundredth Monkey Effect	knowledge_threshold	10
	Threshold_value	1e-35
Genetic	Mutation	0.8
	Crossover	0.2
Simulated Annealing	finalTemp	1e-3
	Alpha	0.95
Bee Colony	limit	50
	LB	-5
	UB	5

B. Simulation result

The simulation results are presented in four figures. Figure 2 corresponds to the Hundredth Monkey Effect algorithm, Figure 3 represents the GA, Figure 4 shows the performance of the SA algorithm, and Figure 5 illustrates the BCO algorithm. These results reveal that the Hundredth Monkey Effect algorithm demonstrates superior convergence performance in solving the Least Squares Minimization Problem compared to the other algorithms. Each algorithm was allowed 200 iterations, with the Hundredth Monkey Effect algorithm in Figure 1 reaching optimal convergence more quickly. Following the Hundredth Monkey Effect, Figure 2 GA shows the best performance, converging faster than the other methods. Figure 4 (BCO) ranks third, and finally, Figure 3 (SA) demonstrates the slowest convergence. This performance comparison confirms that the proposed Hundredth Monkey Effect method converges faster, particularly with fewer iterations, making it more effective for solving the Least Squares Minimization Problem.

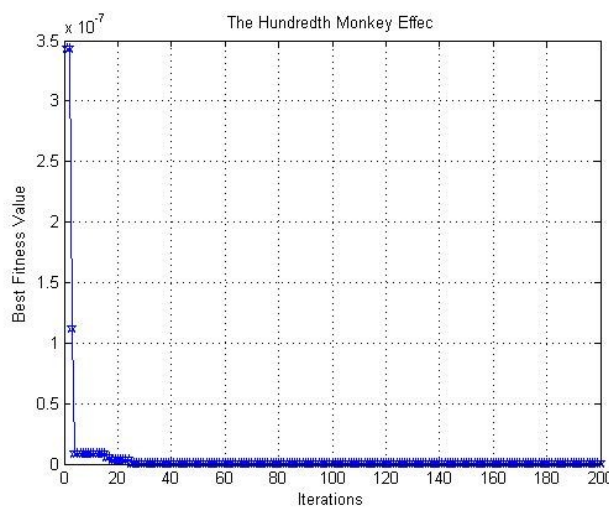


Fig2. Hundredth Monkey Effect.

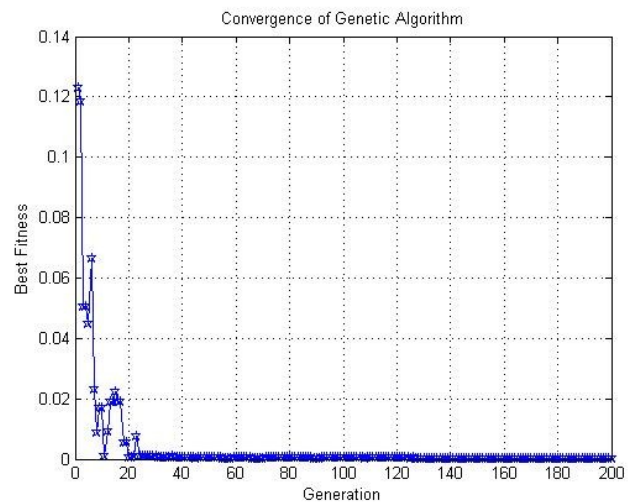


Fig3. Genetic Algorithm.

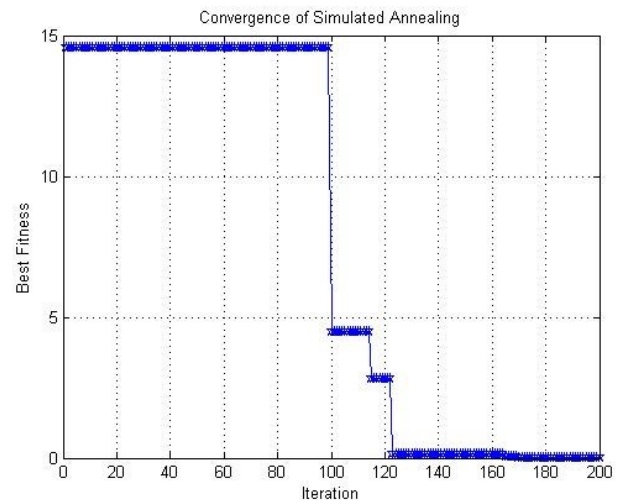


Fig4. Simulated Annealing.

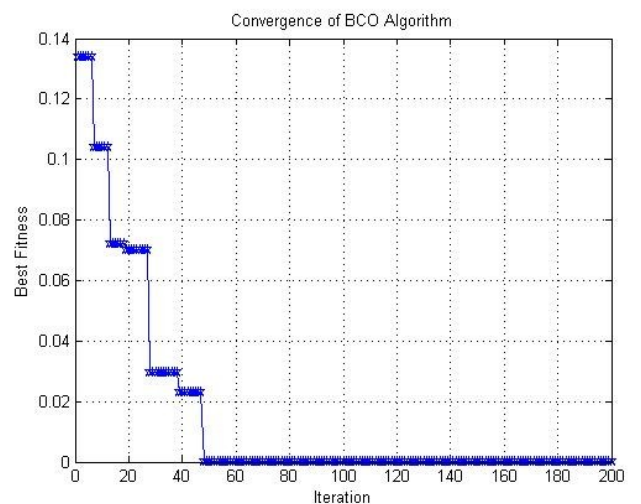


Fig 5. Bee Colony.

V. CONCLUSION

This paper introduces the Hundredth Monkey Effect algorithm as a novel metaheuristic framework for enhancing convergence speed and optimization efficiency, particularly for solving the Least Squares Minimization Problem. Comprehensive simulation results reveal that the proposed algorithm exhibits superior convergence behavior compared to classical approaches, including the GA, SA, and BCO. The proposed algorithm consistently achieves optimal or near-optimal solutions with fewer iterations and lower computational cost, underscoring its robustness and efficiency in addressing complex, large-scale, and time-sensitive optimization problems. Furthermore, the adaptive learning strategy embedded within the popose algorithm effectively balances exploration and exploitation, enabling it to escape local optima and sustain a strong global search capability. Owing to these characteristics, the proposed algorithm demonstrates considerable potential as a powerful and scalable optimization tool for a wide range of real-world engineering, computational intelligence, and data-driven applications.

REFERENCES

- [1] Y. Salami, "SO-ITS: a secure offloading scheme for intelligent transportation systems in federated fog-cloud," *Iran J. Comput. Sci.*, 2025, doi: 10.1007/s42044-025-00318-9.
- [2] Y. Salami, "SOBT-UF: Secure Offloading in Blockchain Infrastructure for Intelligent Transportation Systems Using 5G-Enabled UAVs Within a Fog-Edge Computing Federation," in *2024 19th Iranian Conference on Intelligent Systems (ICIS)*, IEEE, 2024, pp. 217–222. doi: 10.1109/ICIS64839.2024.10887460.
- [3] Y. Salami, F. Taherkhani, Y. Ebazadeh, M. Nemati, V. Khajehvand, and E. Zeinali, "Blockchain-Based Internet of Vehicles in Green Smart City: Applications and Challenges and Solutions," *Anthropog. Pollut.*, vol. 7, no. 1, pp. 87–96, 2023, doi: 10.22034/AP.2023.1978624.1144.
- [4] Y. Salami and S. Hosseini, "BSAMS: Blockchain-Based Secure Authentication Scheme in Meteorological Systems," *Nivar*, vol. 47, no. 120–121, pp. 181–197, 2023, doi: <https://doi.org/10.30467/nivar.2023.415722.1260>.
- [5] Y. Salami, V. Khajehvand, and E. Zeinali, "LSMAK-IOV: Lightweight Secure Mutual AKE Scheme in Fog-Based IoV," in *2024 10th International Conference on Artificial Intelligence and Robotics (QICAR)*, IEEE, 2024, pp. 1–5. doi: 10.1109/QICAR61538.2024.10496659.
- [6] Z.-H. Zhan, L. Shi, K. C. Tan, and J. Zhang, "A survey on evolutionary computation for complex continuous optimization," *Artif. Intell. Rev.*, vol. 55, no. 1, pp. 59–110, 2022.
- [7] B. Akay, D. Karaboga, and R. Akay, "A comprehensive survey on optimizing deep learning models by metaheuristics," *Artif. Intell. Rev.*, vol. 55, no. 2, pp. 829–894, 2022.
- [8] K. Rajwar, K. Deep, and S. Das, "An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges," *Artif. Intell. Rev.*, vol. 56, no. 11, pp. 13187–13257, 2023.
- [9] T. Dokeroglu, E. Sevinc, T. Kucukyilmaz, and A. Cosar, "A survey on new generation metaheuristic algorithms," *Comput. Ind. Eng.*, vol. 137, p. 106040, 2019.
- [10] A. H. Gandomi, X.-S. Yang, S. Talatahari, and A. H. Alavi, "Metaheuristic algorithms in modeling and optimization," *Metaheuristic Appl. Struct. infrastructures*, vol. 1, pp. 1–24, 2013.
- [11] D. Henderson, S. H. Jacobson, and A. W. Johnson, "The theory and practice of simulated annealing," *Handb. metaheuristics*, pp. 287–319, 2003.
- [12] A. Hassanat, K. Almohammadi, E. Alkafaween, E. Abunawas, A. Hammouri, and V. B. S. Prasath, "Choosing mutation and crossover ratios for genetic algorithms—a review with a new dynamic approach," *Information*, vol. 10, no. 12, p. 390, 2019.
- [13] N. Bacanin, M. Antonijevic, T. Bezdán, M. Zivkovic, K. Venkatachalam, and S. Malebary, "Energy efficient offloading mechanism using particle swarm optimization in 5G enabled edge nodes," *Cluster Comput.*, vol. 26, no. 1, pp. 587–598, 2023.
- [14] Y. Salami, "Dialectic optimization algorithm (DOA): a novel metaheuristic inspired by dialectical philosophy," *J. Supercomput.*, vol. 81, no. 15, 2025, doi: 10.1007/s11227-025-07879-3.
- [15] M. Abdel-Basset, L. Abdel-Fatah, and A. K. Sangaiah, "Metaheuristic algorithms: A comprehensive review," *Comput. Intell. Multimed. big data cloud with Eng. Appl.*, pp. 185–231, 2018.
- [16] H. Dahrouj *et al.*, "An overview of machine learning-based techniques for solving optimization problems in communications and signal processing," *IEEE Access*, vol. 9, pp. 74908–74938, 2021.
- [17] V. Kesavan, R. Kamalakannan, R. Sudhakarapandian, and P. Sivakumar, "Heuristic and meta-heuristic algorithms for solving medium and large scale sized cellular manufacturing system NP-hard problems: A comprehensive review," *Mater. today Proc.*, vol. 21, pp. 66–72, 2020.
- [18] S. Desale, A. Rasool, S. Andhale, and P. Rane, "Heuristic and meta-heuristic algorithms and their relevance to the real world: a survey," *Int. J. Comput. Eng. Res. Trends*, vol. 351, no. 5, pp. 2349–7084, 2015.
- [19] H. S. Yahia *et al.*, "Comprehensive Survey for Cloud Computing Based Nature-Inspired Algorithms Optimization Scheduling," *Asian J. Res. Comput. Sci.*, vol. 8, no. 2, pp. 1–16, 2021, doi: 10.9734/ajrcos/2021/v8i230195.
- [20] A. Acheampong, Y. Zhang, X. Xu, and D. A. Kumah, "A review of the current task offloading algorithms, strategies and approach in edge computing systems," *Comput. Model. Eng. Sci. 134*, pp. 35–88, 2023.
- [21] M. Adhikari, S. N. Srirama, and T. Amgoth, "A comprehensive survey on nature-inspired algorithms and their applications in edge computing: Challenges and future directions," *Softw. Pract. Exp.*, vol. 52, no. 4, pp. 1004–1034, 2022.
- [22] M. Abd Elaziz *et al.*, "Advanced metaheuristic optimization techniques in applications of deep neural networks: a review," *Neural Comput. Appl.*, pp. 1–21, 2021.
- [23] A. Yeredor, "The extended least squares criterion: Minimization algorithms and applications," *IEEE Trans. signal Process.*, vol. 49, no. 1, pp. 74–86, 2001.
- [24] J. Sjoberg and M. Viberg, "Separable non-linear least-squares minimization-possible improvements for neural net fitting," in *Neural networks for signal processing VII. Proceedings of the 1997 IEEE signal processing society workshop*, IEEE, 1997, pp. 345–354.
- [25] W. Hoitinga, R. de Groot, M. Kwakkel, and M. Gerritsma, "Direct Minimization of the least-squares spectral element functional—Part I: Direct solver," *J. Comput. Phys.*, vol. 227, no. 4, pp. 2411–2429, 2008.
- [26] S. Amaran, N. V. Sahinidis, B. Sharda, and S. J. Bury, "Simulation optimization: a review of algorithms and applications," *Ann. Oper. Res.*, vol. 240, pp. 351–380, 2016.