Universiteit Leiden

Master Computer Science

Optimizing office lighting in terms of energy and uniformity based on presence detection

Name:	Katerina Zacharia
Student ID:	3783049
Date:	09/07/2025
Specialisation:	Artificial Intelligence
1st supervisor:	Niki van Stein
2nd supervisor:	Anna Kononova

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Contents

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Dr. Niki van Stein, for her exceptional and continuous guidance, and insightful feedback throughout the course of this thesis. Her expertise and encouragement have been invaluable in shaping the direction of this research and ensuring its successful completion. I am particularly grateful for the time and effort she dedicated to helping me navigate both conceptual and practical challenges with clarity and confidence.

I would also like to sincerely thank my sister, Michaella Zacharia, a lighting design expert, for her technical support and assistance with the DIALux simulator. Her help was crucial in the implementation phase of the project, especially during the more demanding stages of simulation integration and testing.

I am truly thankful to my family and boyfriend for their unwavering support, patience, and encouragement during my academic journey. Their belief in my work and constant motivation provided me with the strength to persevere through moments of difficulty and uncertainty.

Furthermore, I would like to acknowledge the contributions of my fellow students and peers, whose suggestions and shared experiences inspired new ideas and perspectives that significantly enriched the development of this work.

Abstract

In recent years, the demand for energy-efficient and occupant-friendly lighting systems has grown significantly, particularly in commercial buildings where lighting constitutes a substantial portion of energy consumption. This thesis proposes a black-box optimization framework for smart indoor lighting control, aiming to minimize energy usage while ensuring user comfort through adequate illuminance and uniformity. The problem is modeled as both a Single-Objective and a Multi-Objective optimization task, with constraints based on European lighting standards (EN 12464-1). Due to the simulation-based and non-differentiable nature of the evaluation process, executed via the DIALux evo software, traditional optimization techniques are unsuitable. Therefore, this study explores non-traditional black-box methods, specifically two variants of Bayesian Optimization (BO) and Particle Swarm Optimization (PSO). The proposed system adapts lighting configurations based on randomized sensor input reflecting presence/motion, covering a wide range of occupancy scenarios. The algorithms were tested on eleven discrete sensor configurations, and evaluated in terms of energy consumption, desks illuminance, and lighting uniformity. Results show that Bayesian Optimization techniques, particularly when warm-started and with the Probability of Improvement acquisition function, outperform PSO in most scenarios, achieving a better balance between efficiency and comfort, and indicate the fastest convergence. PSO certainly needs either more evaluations for this kind of problem or special treatment, as we will discuss. In the Multi-Objective setting, SAMO-COBRA achieves superior Pareto front quality, effectively balancing energy and uniformity when provided with a sufficient evaluation budget. This research highlights the suitability of black-box optimization for smart lighting control, and provides a comparative analysis of algorithm performance across practical conditions, contributing to the development of adaptive and energy-conscious indoor environments.

1 Introduction

The increasing demand for energy-efficient technologies in commercial buildings is driven by rising electricity costs, growing global consumption, and heightened environmental concerns [7] [21] [3]. Among building operations, indoor lighting remains a major contributor to energy use, particularly in office environments. Despite advancements in lighting technology, many systems operate inefficiently, often remaining active throughout the day regardless of occupancy levels or daylight availability. This leads to excessive energy consumption and elevated operational costs. At the same time, lighting quality plays a crucial role in occupant well-being [30] [1]. Inadequate lighting design can cause discomfort, eye strain, fatigue, and reduced productivity. These challenges have intensified interest in smart lighting systems that strive to balance energy efficiency with user comfort; two objectives that frequently conflict in practice.

This thesis addresses this dual objective by developing an intelligent lighting control system that adapts illumination based on occupancy data. The problem is framed as a black-box optimization task, acknowledging the simulation-based nature of lighting evaluation in tools like DIALux evo [12], which do not provide analytical expressions for key performance metrics such as energy consumption or uniformity.

To solve this problem, we investigate multiple optimization techniques, including two variants of Bayesian Optimization and Particle Swarm Optimization, applied to a series of randomized occupancy scenarios. System performance is evaluated using three core metrics aligned with European lighting standards (EN 12464-1) [10], that are energy consumption, desks illuminance, and lighting uniformity.

By applying black-box optimization in a simulation-driven context, this work advances the development of adaptive lighting strategies for smart buildings. It offers insights into algorithm performance and highlights how such systems can achieve sustainability objectives without compromising user comfort.

1.1 Problem Statement

Optimizing indoor office lighting requires balancing two competing objectives: minimizing energy consumption and maintaining or even maximizing user comfort. This becomes more complex under dynamic occupancy conditions and regulatory lighting standards, which limit the effectiveness of traditional rule-based or model-driven approaches that rely on explicit, differentiable models.

In this thesis, lighting configurations are evaluated through DIALux evo, which operates as a black-box simulator for our system, making the internal mechanics inaccessible and each function evaluation computationally expensive. Additionally, such systems must respond adaptively to real-time input from motion/presence sensors, further complicating the search for optimal lighting settings. In this project, we simplify by removing the real-time insertion of occupancy, and instead simulate some predefined scenarios.

The core problem, therefore, is to determine efficient lighting configurations, defined by lamp-specific lumen values, under varying occupancy scenarios, while satisfying constraints on illuminance and uniformity and minimizing energy consumption. Solving this requires sample-efficient optimization techniques capable of operating in a black-box, constraint-driven environment.

1.2 Goals of the Thesis

As discussed, the primary goal of this thesis is to design an intelligent lighting control system that balances our two objectives using modern optimization methods and sensor-based data. The specific objectives are:

- **To evaluate the extent to which an intelligent lighting control system can extract optimal lighting configurations for commercial office spaces while minimizing manual intervention.** The system should operate autonomously, enhancing user convenience and enabling seamless adaptation to occupancy conditions.
- **To quantify the impact of energy-efficient lighting optimization on reducing operational costs in office environments.** Energy consumption has both financial and environmental implications. Reducing it lowers electricity expenses and CO_2 emissions, contributing to sustainability and corporate responsibility.
- **To examine the role of user comfort in lighting systems and assess the system's ability to adapt lighting behavior to human presence.** While energy reduction is essential, the system must also support visual comfort and productivity by avoiding unnecessary dimming or switching and by maintaining a suitable lighting environment.
- **To assess the effectiveness of sensor-driven data in enabling adaptive lighting responses to varying occupancy scenarios.** Leveraging data from motion/presence sensors, the system should intelligently adjust light levels according to space utilization, ensuring efficiency without sacrificing usability.

The system will utilize sensor data, particularly motion/presence detection, to determine the optimal lumens and hence wattage for each lamp in the office environment. Since this is an optimization problem, certain constraints must be satisfied to ensure the feasibility and effectiveness of the proposed solution. According to the European Commission regulations (EN 12464-1:2021 [10]) , the minimum required illuminance for office workstations should be at least 500 lux to ensure proper visibility and comfort. Additionally, the illumination uniformity across the office space must exceed a predefined threshold to prevent discomfort caused by glare or uneven lighting. To achieve this, we implement both Single- and Multi-Objective black-box optimization frameworks that adjust lighting configurations based on sensor and lighting data.
We aim to compare Bayesian Single- and Multi-Objective optimization methods against the commonly used PSO from previous research [31] [17].
For the correct evaluation of the solutions, we utilize DIALux evo, a lighting simulation software capable of accurately modeling illuminance levels, energy consumption, and uniformity distribution in indoor environments. By simulating the lighting conditions under different optimization-generated configurations, DIALux enables us to assess whether the system meets both energy efficiency goals and regulatory lighting standards (e.g., maintaining at least 500 lux per workstation).
Summarizing, the primary research question is: Do Single- and/or Multi-Objective Bayesian Optimization methods achieve greater efficiency in optimizing energy consumption and illumination uniformity in indoor lighting systems, based on sensor data, compared to the widely used PSO algorithm?

The structure of this paper is organized as follows. Section 2 presents a review of related work that frames the context of this study. Section 3 outlines the theoretical background and methodology that underpin the approach taken. Section 4 details the implementation process, followed by Section 5 which describes the experimental setup. Section 6 reports the results obtained, while Section 7 provides a discussion and interpretation of these findings. Finally, Section 8 concludes the paper and suggests directions for future research.

2 Related work

There has been a satisfactory amount of research on smart lighting, whether to save energy or to achieve a better user experience [24][6][5]. There is also research on how machine learning methods can be integrated into such systems, or into IoT in general [22] [13] [23]. First, we are going to present research that examined rule-based techniques, followed by meta-heuristic methods. Then, BO and Multi-Objective algorithms are discussed in order to investigate their contribution in this field.

A comprehensive literature review conducted by A. G. Putrada et al. [22] explores the evolution of smart lighting systems since 2014, with particular emphasis on machine learning applications and user comfort considerations. Their analysis covers fundamental challenges in smart lighting, including energy consumption, control systems, and network infrastructure, while examining implementations across various contexts from homes to offices. The review details essential technological components such as LEDs, sensors, actuators, and IoT integration, highlighting how machine learning enhances smart lighting capabilities through predictive control, intelligent sensor-based actuation, gesture recognition, and activity detection. Notably, the authors provide detailed frameworks for quantifying user comfort metrics, which are crucial for evaluating system performance. The review concludes by addressing key development challenges, including cost-effectiveness and optimal sensor selection, providing valuable insights for future smart lighting implementations.

Martirano [18] proposed an energy-efficient lighting control framework based on actual building usage patterns and user requirements. The approach focuses on minimizing unnecessary energy consumption by monitoring daylight availability and occupancy levels across different building zones, with a case study implemented in a classroom setting. The research presents a supervisory control system framework utilizing sensor networks for data collection. While the study demonstrates potential for energy savings through strategic control mechanisms, it lacks the implementation of specific AI methods and comprehensive data analysis. Although this framework could serve as a baseline for smart lighting control systems, it notably omits consideration of user comfort metrics, highlighting a gap between energy optimization and occupant satisfaction that warrants further investigation. We contribute to this work by using advanced optimization techniques by taking into account a more human-centric approach.

The study [29] investigates energy conservation through smart lighting in offices, emphasizing visual comfort and energy savings. It proposes two primary energy-saving methods: occupancy-based ON/OFF switching and conditional dimming, tailored to adapt across diverse office environments. Instead of AI, these systems rely on predefined logic, utilizing IoT sensors to enhance energy efficiency based on occupant presence. The study compares these smart features against a baseline scenario, highlighting substantial energy savings while ensuring visual comfort. Evaluation is primarily statistical, focusing on quantifiable energy and occupancy metrics to assess impact effectively. The lack of this work comes from the fact that it is a rule-based method with predefined conditions, instead of integrating smarter methods, as we are intending to undertake.

The article by Ayan and Turkay [2] examines the impact of smart LED bulbs in IoT-enabled lighting systems on energy efficiency, focusing on how various colors affect power consumption. It performs an extensive analysis on types of LEDs, examining both the advantages of smart lighting and how different colors contribute to energy use. The study's findings support the practical benefits of smart lighting within home automation, including potential reductions in energy consumption. However, it does not cover elements such as user comfort or spe-

cific lighting control methods like ON/OFF or dimming functions, instead evaluating energy consumption in various configurations to determine overall efficiency.
Another employment explores Meta-heuristic methods [28]. This research considers residential buildings prioritizing HVAC systems but also considers lighting as part of an IoT-based setup. The study applies algorithms, including Sine Cosine (SCA), Modified Sine Cosine (MSCA), Arithmetic Optimization (AOA), and Hill-Climbing (HC), to optimize energy use. Additionally, it considers daylighting effects but lacks specific attention to user comfort, thus focusing mainly on energy savings in sunny conditions without broader environmental variability. Previous research by these authors [27] applies multiple optimization algorithms, including Particle Swarm Optimization and Genetic Algorithm, to refine both energy efficiency and user comfort in home energy management. While the aforementioned study focused primarily on cost savings in HVAC and lighting, this work introduces additional optimization techniques and evaluates the system under specific comfort constraints, such as maintaining indoor temperatures between 23-26°C. This paper gives more attention to occupant comfort alongside efficiency, utilizing YALMIP for solving optimization problems. While comfort is considered, lighting is not the main focus, and sensor integration is minimal. This thesis focuses directly on lighting, using realistic sensor inputs to drive decisions and evaluates PSO's performance.
Latest research has started to explore the potential of Bayesian Optimization in building energy management. Lin et al. [15] crafted a Bayesian Optimization approach for HVAC control systems and demonstrated it to be able to optimize energy consumption as well as thermal comfort. They highlight the capability of BO in handling real-world, black-box constraint-based optimization problems in building environments. Similarly, Xu et al. [26] propose a Primal–Dual Contextual Bayesian Optimization (PDCBO) framework for HVAC control under dynamic daily constraints. The study emphasizes BO's strength in handling Multi-Objective and contextual optimization, making it a compelling alternative to conventional controllers when occupant comfort and energy efficiency must be simultaneously addressed. Both researches showcase that BO was applied mostly in the HVAC problem instead of lighting. Our work, though, applies BO to smart lighting, showing its applicability to a new domain with similar black-box and constraint-heavy characteristics.
Multi-Objective methods were also employed in the following studies. Madias et al. [16] explore an optimization model for interior lighting using genetic algorithms, NSGA-II specifically. It aims to minimize energy consumption while maximizing uniformity and maintaining adequate illumination levels. The study employs simulated data and focuses on two objective functions: the total dimming levels of luminaires and the coefficient of variation of illuminance uniformity. While achieving significant energy savings, the model does not utilize daylighting data or occupancy sensors, limiting its focus on user comfort. Its adaptable design allows application across various room configurations. In contrast, Wagiman et al. [25] present a more recent approach to lighting optimization with similar goals. They introduce the Illuminance Uniformity Deviation Index (IUDI), a metric for measuring illuminance uniformity, which is optimized alongside power demand using Multi-Objective Particle Swarm Optimization (MOPSO). This approach allows for the simultaneous consideration of both artificial lighting and daylight, resulting in improved outcomes for energy consumption and visual comfort. This project advances these papers by incorporating occupant-specific comfort constraints in a Multi-Objective framework.
The study by Minh Hoang Ngo et al. [20] explores an adaptive smart lighting control system utilizing a Multi-Objective Genetic Algorithm (GA) framework. The GA aims to optimize two primary objectives: the total energy consumption of the lighting in the room and a penalty function reflecting the difference between user-desired and actual illuminance levels. This sys-

tem enables users to specify their lighting preferences through a mobile application, allowing for the optimization of energy consumption based on these preferences. While it adapts to user needs, the focus is more on optimizing energy use rather than explicitly maximizing overall user comfort. The design incorporates considerations for daylighting and is versatile enough to accommodate various room structures, making it both generalized and adaptable. The researchers employed a simulator to validate the system's performance, demonstrating its capability to manage lighting conditions in real-time based on user input and environmental changes. In contrast, we aim to rely on sensor inputs and occupancy rather than user-declared preferences.

To build upon existing research in smart lighting, this study develops and evaluates three optimization algorithms, both Single- and Multi-Objective, aimed at achieving a balance between energy efficiency and user comfort. The goal is to compare the performance of multiple algorithms and problem formulations using occupancy data. While prior work has explored energy-saving strategies or user-oriented lighting separately, few studies use black-box optimization methods to address both objectives simultaneously. In particular, Bayesian Optimization remains underexplored in the context of smart lighting, despite its strength in handling expensive, non-differentiable, and Multi-Objective problems. Most existing approaches rely on rule-based logic or traditional metaheuristics without leveraging surrogate modeling. This thesis fills this gap by formulating the problem as a black-box optimization task, integrating occupancy-based sensor data, and comparing the performance of two BO variants against PSO.

3 Methodology

In this chapter, the mathematical formulation of the problem will be presented and discussed. Furthermore, the algorithms used in this project will be analyzed by also providing their strengths and weaknesses.

3.1 Problem formulation

In optimization problems, the aim is to find the best set of input values that minimize or maximize an objective function. In black-box optimization, the internal structure of this function is unknown or inaccessible; we can only observe the input–output behavior of the system.
Formally, given an unknown objective function f: $\mathbb{R}^n \rightarrow \mathbb{R}$, a black-box optimization problem can be defined as: $\min_{x \in \mathcal{X}} f(x)$. Here, x is a vector of decision variables, $\mathcal{X}$ is the feasible domain defined by constraints, and $f(x)$ is the unknown function whose value can only be obtained through costly evaluations.
Traditional methods typically include manually defined rules or logic-based structures such as condition statements, as well as classical optimization techniques that require an explicit and often differentiable objective function. However, the problem addressed in this project lacks a closed-form objective and depends on simulation-based evaluations. These characteristics make it unsuitable for traditional approaches and instead require the use of non-traditional (black-box) optimization methods, which can operate without direct knowledge of the function's structure and are effective for complex, costly, and data-driven problems.
The specific properties of this project that justify the use of black-box optimization techniques are:

- Multi-Objective with trade-offs: The problem requires balancing energy consumption and user comfort, specifically uniformity.
- Expensive to evaluate: Each candidate solution must be assessed through a time-consuming DIALux simulation, making sample efficiency critical.
- Black-box by nature: The system does not provide an explicit analytical relationship between inputs (e.g., lumens) and outputs (e.g., lux, energy, and uniformity), making traditional methods impractical.

Because of these settings, optimization methods such as gradient descent or convex optimization are not applicable. Black-box optimization techniques are better suited for this kind of problem [11].
In this context, there are two decision variables:

- L_i representing lumens of lamp i, where $L_i \epsilon [0, L_{max}]$ and $L_{max} = 3600$.
- W_i illustrate the wattage of lamp i, which is defined as $W_i = \frac{L_i}{L_{eff}}$. L_{eff} is the luminous efficacy of a lamp, this is a standard number for each type and model; the specific lamp we used has $L_{eff} = 124.1$.

The problem can be formulated as either a Single-Objective or a Bi-Objective optimization task, depending on the priorities set between energy efficiency and lighting quality. In the following sections, we present both formulations and explain how the objective function(s) are defined accordingly.

3.1.1 Single-Objective vs Bi-Objective

For the cases where we apply just one objective function, we set a balanced sum of energy consumption and negated uniformity, formulated as:

$$f(L) = \frac{\sum_{i=1}^{N} W_i}{E_{max}} - U + 0.1 \times Penalty \tag{1}$$

where f is the objective function, the fraction represents the normalized energy (the total maximum energy is defined by the number of lamps multiplied by the maximum watts, which is 29), and U is the uniformity and measures how well the light is distributed among the area we examine. The $Penalty$ will be explained in Section 3.1.2.
In the Bi-Objective case, we have two objective functions defined as:

$$min \sum_{i=1}^{N} W_i \tag{2}$$

where N is the number of lamps, and the sum is the total power consumption for all the lamps for a specific moment.

$$maxU \tag{3}$$

where U is the uniformity of the area calculated. It is important to mention that uniformity concerns the whole area and not specific surfaces or spots in the place we investigate.

3.1.2 Penalties/Constraints

The two main constraints are Equation 4 and 5. Respectively, uniformity U of a place should be above 0.6 to consider the lighting as ideal, and the lux levels for each of the desks (workstations) should exceed the threshold of 500 lux.

$$U \geq 0.60 \tag{4}$$

$$E_{workstation,j} \geq 500, \forall j \tag{5}$$

The *Penalty* term in Equation 1 is designed to quantify deviations from ideal lighting conditions, and it consists of two components: uniformity and desk illuminance penalty. For any given configuration, the uniformity penalty is computed as the difference between the ideal U (e.g., 0.6) and the actual one obtained from the simulation. This difference is multiplied by 100 to scale it in the same range as the lux-based penalty. The desk illuminance penalty measures how much the current desk lux deviates from the target of 500 lux, calculated as the absolute difference between 500 and the actual lux value. For example, if the room's U is 0.5 and the examining desk illuminance is 400 lux, the $Penalty$ is computed as: $(0.6 - 0.5) \times 100 + (500 - 400) = 10 + 100 = 110$.
However, before applying these penalties, presence/motion detection is taken into account. If no activity is detected in a zone, the uniformity penalty is ignored for that zone, as poor U in unoccupied areas is not problematic unless it drastically affects the general U. Regarding the desk penalty, it is represented by a slightly different definition. If a desk is not occupied, the penalty applies only when the lux exceeds 500, indicating wasted energy. For instance, if a vacant desk shows 550 lux, the penalty is 50. Otherwise, i.e., if a desk is occupied, the

penalty is applied as explained in the previous paragraph. So, for an occupied desk with only 450 lux, the penalty is again 50. This penalty formulation ensures that both under-illumination for occupants and over-illumination for empty desks are penalized, while unnecessary penalties for uniformity in unoccupancy states are avoided.
In the Bi-Objective manner, the total $Penalty$, defined as the desk lux penalty added to the uniformity penalty, acts as a single constraint. It follows the same logic as explained previously.

3.2 Algorithms

As mentioned earlier, this thesis employs both Single- and Bi-Objective optimization techniques. Plain Bayesian Optimization and Particle Swarm Optimization (PSO) are implemented as Single-Objective methods, as described in Equation 1, while the SAMO-COBRA framework is used to explore the problem in a Bi-Objective setting.
The choice to focus on two BO variants and PSO is motivated by both their technical relevance and research significance. Although BO has gained traction in fields such as machine learning and engineering design, it remains underexplored in the context of smart lighting control. Most existing work in this area relies on rule-based logic or metaheuristics with limited investigation into BO's potential for sample-efficient optimization in simulation-driven environments. PSO, on the other hand, is a well-established population-based algorithm that serves as a robust baseline due to its simplicity, flexibility, and proven success in energy-related applications.
The primary objective of incorporating BO is to evaluate its effectiveness in addressing the black-box, constraint-driven nature of the indoor lighting problem. In particular, the study investigates BO's ability to produce high-quality configurations, those that minimize energy consumption while satisfying comfort and uniformity constraints, with significantly fewer function evaluations compared to other approaches. This is especially relevant given the high computational cost associated with each simulation run. A secondary objective is to assess the practical applicability and robustness of BO across different configurations, including variations in initial sample sizes and acquisition functions. By systematically benchmarking BO against PSO, the study contributes not only performance insights but also a critical evaluation of whether BO offers a viable and scalable solution for adaptive lighting systems in smart lighting environments.

3.2.1 Plain Bayesian Optimization

Bayesian Optimization is a sequential, model-based optimization technique designed for expensive black-box functions. Unlike gradient-based methods, BO builds a probabilistic model of the objective function and iteratively selects the most promising configurations to evaluate. In this thesis, BO is used to optimize the lighting configurations, determining the optimal lumens and thus wattage for each lamp based on sensor data.
Bayesian Optimization follows an iterative process, as depicted in the below list.

1. Start with an initial set of sample points from the objective function

2. Build a surrogate model by fitting a probabilistic model (usually a Gaussian process) to approximate the objective function

3. Use an acquisition function to decide which point to evaluate next

4. Evaluate the actual objective function (e.g., energy consumption and uniformity from DIALux)

5. Update the surrogate model by incorporating the new data

6. Repeat the procedure

This approach allows BO to balance exploration (trying new areas) and exploitation (focusing on known good areas) efficiently. A Gaussian process (GP) is used to model the unknown objective function. Given a set of past evaluations, the posterior distribution of $f(x)$ is given by Equation 6.

$$f(x) \sim GP(\mu(x), \kappa(x, x')) \tag{6}$$

where μ is the mean function (expected function value), and $\kappa(x, x')$ is the covariance kernel, which defines similarity between points. The purpose of GP is to estimate both mean function values and uncertainty, guiding the next evaluation step.
To decide where to sample next, an acquisition function is used. Some popular options include the Expected Improvement (EI), Upper Confidence Bound (UCB), and Probability of Improvement (PI). EI prioritizes points that are likely to improve upon the best observed value, whilst UCB balances exploration and exploitation using a variable κ to decide the trade-off between these two. PI is able to choose points with the highest probability of improving the best-found solution.
In this paragraph, we will mention some noteworthy advantages and disadvantages of Bayesian Optimization. BO is very efficient when it comes to expensive evaluations, such as the DIALux simulations. Moreover, it is able to handle uncertainty through the Gaussian processes. However, as with all algorithms, BO comes with its limitations, which are the high computational cost, especially with large datasets. Furthermore, it is essential to provide good initial points, making it sometimes necessary to set a warm-start. Finally, deciding the acquisition function is important since the different options (EI, UCB and PI) impact the performance.

3.2.2 SAMO-COBRA

SAMO-COBRA (Surrogate-Assisted Self-Adaptive Multi-Objective Constrained Optimization by Radial Basis Approximation) is an advanced surrogate-assisted black-box optimization technique designed for Multi-Objective constrained problems. It is particularly useful when function evaluations are expensive, such as DIALux simulations for lighting control.
Unlike previous BO, which relies on Gaussian processes, SAMO-COBRA uses Radial Basis Function (RBF) models as a surrogate to approximate the objective function and constraints. This enables faster convergence and better handling of constraints compared to PSO. SAMO-COBRA follows an iterative process similar to the aforementioned Bayesian Optimization but with a different surrogate modeling approach:

1. Initialize with sample points using Latin Hypercube Sampling (LHS) or random sampling

2. Build a Radial Basis Function (RBF) surrogate model to approximate the objective function based on previously evaluated solutions

3. Identify the next promising configuration using a balance of exploration and exploitation

4. Evaluate the actual objective function

5. Update the surrogate model by incorporating new results into the RBF model and refining predictions

6. Repeat until convergence or a stopping criterion is met

Some benefits of this algorithm are that it is more sample-efficient compared to PSO, as it utilizes fewer function evaluations than heuristic methods. The algorithm is designed to manage constraints explicitly and adaptively. In this project, constraints such as achieving at least 500 lux at each desk and maintaining minimum uniformity are critical. SAMO-COBRA models and refines these constraints, enabling it to find feasible solutions more effectively than traditional methods. It automatically selects and adjusts surrogate model parameters throughout the optimization process. However, there are also several limitations. Although faster than direct evaluations, RBF models can become computationally expensive as more samples are added. SAMO-COBRA may overfit to known solutions and struggle with high-dimensional search spaces. Finally, it requires careful surrogate model selection since RBF may not always be the best model.

3.2.3 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is an optimization algorithm inspired by real nature, as it relies on the collective behavior of swarms (such as birds or fish) searching for food. It is a population-based, iterative method that optimizes a problem by iteratively improving candidate solutions based on their position and velocity in the search space.
More in detail, PSO is based on a swarm of particles, where each particle represents a candidate solution. The swarm collectively searches for the optimal solution by adjusting their positions based on the best-known solutions found so far. Each particle has:

- Position (X_i) that represents a possible lighting configuration
- Velocity (V_i) which decides how the position changes in the next iteration
- Personal best position (P_i) is the best solution found by the particle so far, and
- Global best position (G) illustrates the best solution found by any particle in the swarm

At each iteration, particles update their velocity and position using the following equations:

1. Velocity Update Equation (7), where: V_i^t is the velocity of particle i at iteration t, ω is the inertia weigh (how much of the previous velocity will be retained); c_1 and c_2 represents the acceleration coefficients; r_1 and r_2 are random values between 0 and 1. P_i is the personal best position of the particle and G the global best position found by any particle.

2. Position Update Equation (8), where X_i^t is the particle's current position (i.e., the lighting configuration).

$$V_i^{t+1} = \omega V_i^t + c_1 r_1 (P_i - X_i^t) + c_2 r_2 (G - X_i^t) \tag{7}$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \tag{8}$$

Each particle updates its velocity and position iteratively, balancing exploration (searching new areas) and exploitation (refining good solutions). The steps followed, personalized for our problem, are below:

1. Initialize a swarm of particles, where each particle represents a set of lumens/wattage values for all lamps
2. Evaluate each particle by computing energy consumption and illumination uniformity using DIALux simulations
3. Update the personal and global best solutions based on the evaluation results
4. Adjust velocity and position using the update equations
5. Repeat until convergence (e.g., when improvements become negligible or a maximum number of iterations is reached)

PSO is fast in convergence when compared to other exhaustive search methods. It is an algorithm easy to implement, with only a few parameters to be tuned. Unlike BO and SAMO-COBRA, this one is effective for high-dimensional spaces. On the other hand, there is a high chance of getting stuck in local optima, not guaranteeing a global optimum. Even though the hyperparameters are few, PSO is heavily dependent on those, so pure tuning may lead to bad results.

3.2.4 Warm-Start

During early tests without warm-start initialization, the algorithms often converged slowly or remained stuck in suboptimal regions of the search space. To mitigate this, a warm-start mechanism was introduced using manually derived configurations on the full occupancy scenario and leads to all lamps being set to 3400 lumens. This configuration serves as an informed starting point for the optimization algorithms throughout all scenarios.
The warm-start notably improved performance in BO and SAMO-COBRA, which rely on surrogate models and benefit from informative initial designs. The use of warm-starts resulted in:

- Faster convergence with fewer simulation calls
- Higher-quality solutions in early iterations
- Improved algorithm stability

In summary, the algorithms discussed in this chapter, for the smart lighting optimization problem, are presented in Table 1. The next chapter discusses how these methods were implemented in practice using Python and DIALux.

Table 1: Benefits and Drawbacks of the Optimization Algorithms

Algorithm	Benefits	Drawbacks
BO	Sample-efficient for expensive black-box functions; balances exploration and exploitation via acquisition functions; handles uncertainty through Gaussian Processes.	High computational cost for large datasets; sensitive to initial points (often needs warm-start); performance varies with acquisition function choice.
SAMO-COBRA	Reduces evaluations using surrogate modeling; handles constraints explicitly; adapts surrogate model and kernel selection during optimization.	Surrogate updates grow expensive with many samples; may overfit known solutions; less effective in high-dimensional spaces.
PSO	Fast convergence; simple to implement with few hyperparameters; suitable for high-dimensional problems.	Prone to local optima; lacks built-in constraint handling; performance sensitive to parameter tuning.

4 Implementation

This chapter outlines the procedures and tools used to implement the system described in this thesis. Section 4.1 provides an overview of the workflow, followed by detailed descriptions of the algorithms and the simulation environment using Python and DIALux.

4.1 Workflow

An overview of the project workflow is illustrated in Figure 1. The process begins with the generation of randomized sensor data, representing presence/motion as boolean values (true for occupancy, false for vacancy). These sensor states are used to create a variety of test scenarios, validating that the system performs robustly across different occupancy patterns.
Additionally, lighting configurations specifying tuples of (lamp name, lumens, watts) are used as input for the iterations of each optimization algorithm. These configurations are parsed and applied in DIALux through UI automation. After the simulation, DIALux exports the results, including lux levels, energy consumption, and uniformity, into Excel files. If the predefined stopping criterion is met (in our case, a maximum number of evaluations), the optimization halts; otherwise, the loop will continue.

4.2 Algorithms implementation

The optimization algorithms were implemented using Python 3.13. For each method, existing libraries compatible with this version were used to enable parameter tuning and ease of experimentation. The selected libraries include:

- `scikit-optimize` for BO (version 1) [14]
- `samo-cobra` for surrogate-assisted self-adaptive Multi-Objective optimization/ BO (version 2) [8]

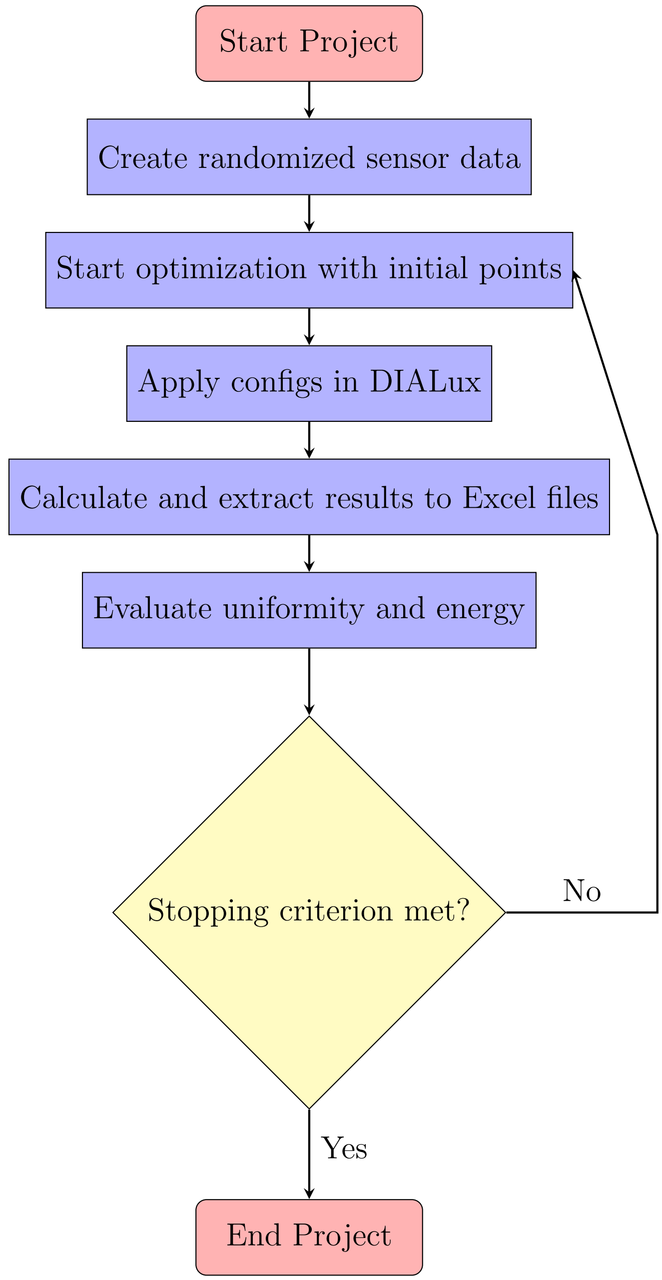

Figure 1: Workflow of the project.

- `pyswarm` for Particle Swarm Optimization [19]

4.2.1 Characteristics

The first library utilized is `scikit-optimize`, a lightweight and flexible Python package designed for sequential model-based optimization. It is particularly effective for hyperparameter tuning and black-box optimization tasks where evaluations are expensive. The optimization process in `scikit-optimize` is driven by surrogate models, such as Gaussian Processes (GP), Random Forests, or Gradient Boosted Trees. In this project, only Gaussian Processes were used. The library supports several acquisition functions, including Expected Improvement (EI), Probability of Improvement (PI), and Lower Confidence Bound (LCB), allowing for a trade-off

between exploration and exploitation.
`SAMO-COBRA`, on the other hand, is a surrogate-assisted Multi-Objective optimizer specifically developed for constrained black-box problems. It uses Radial Basis Function (RBF) models exclusively as surrogate approximators for both objective and constraint functions. The algorithm supports Multi-Objective optimization through non-dominated sorting and maintains an archive of feasible non-dominated solutions, representing the evolving Pareto front. Its iterative search process is designed to balance exploration and exploitation, using surrogate predictions to propose new candidate solutions efficiently while minimizing expensive function evaluations.
Lastly, `pyswarm` is a minimalistic Python library for Particle Swarm Optimization (PSO). It is a population-based optimization method that does not require gradient information, making it suitable for black-box and non-differentiable problems. The implementation allows the user to define bounds and constraints for the search space and supports constraint handling through penalty functions. While it does not include surrogate modeling, `pyswarm` is computationally efficient and simple to configure, making it a practical choice for baseline comparisons.

4.3 DIALux Software

DIALux, developed by DIAL GmbH in Lüdenscheid, Germany, is a powerful lighting design and simulation software widely used by professionals to plan, visualize, and optimize lighting systems for both indoor and outdoor environments. The software offers a comprehensive suite of tools for creating detailed 3D models of spaces, enabling users to simulate lighting conditions with high accuracy. DIALux supports an extensive library of luminaires from various manufacturers, allowing designers to select and integrate specific lighting products into their projects. The software calculates key lighting metrics, such as illuminance (i.e., lux levels), luminance, glare, and uniformity, ensuring compliance with international lighting standards like EN 12464-1. Additionally, DIALux provides advanced features for daylight analysis, energy efficiency evaluations, and dynamic lighting scenarios, making it an indispensable tool for architects, lighting designers, and engineers aiming to achieve optimal lighting solutions that balance performance, energy efficiency, and user comfort.
In this project, DIALux is utilized in the evaluation part of all the optimization algorithms. After the algorithm chooses the new points to evaluate, i.e., the new lumen values and hence wattages for each lamp, they are inserted in DIALux, which provides the option to change lumens and watts per lamp, as seen on the left pane in Figure 2. The next step is to calculate the illuminance levels, uniformity, and energy. The 'Documentation' tab in DIALux provides us with those and more details. An example image of how this looks is given in Figure 3. As we can observe, it shows the final results calculated (with built-in formulas) and whether these exceed the desired threshold, defined again by the program. In this example, it is clear that the average lux level is correctly above 500, uniformity is above 0.60, and energy consumption is within the limits, but the glare valuation should be below or equal to 19, which does not apply.

4.3.1 Setup

We created a simple 3D room (Figure 2) with six working desks and corresponding chairs, ten presence/motion sensors placed above these desks, and 18 lamps; four per sensor and workspace. The room also consists of an entrance door and five windows on one side. The lamps used are CoreLine panel RC133V G4 LED36S/840 PSD W62L62 OC from the Philips

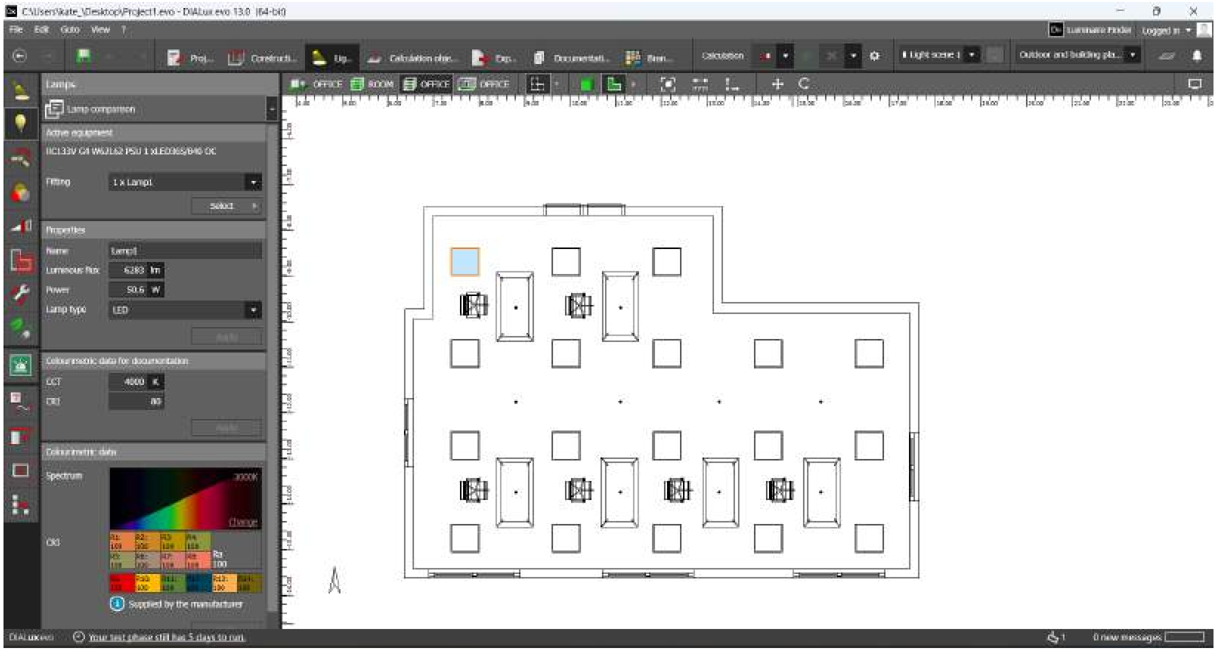

Figure 2: A snapshot from the DIALux software, also showcasing the structure of the room we are examining.

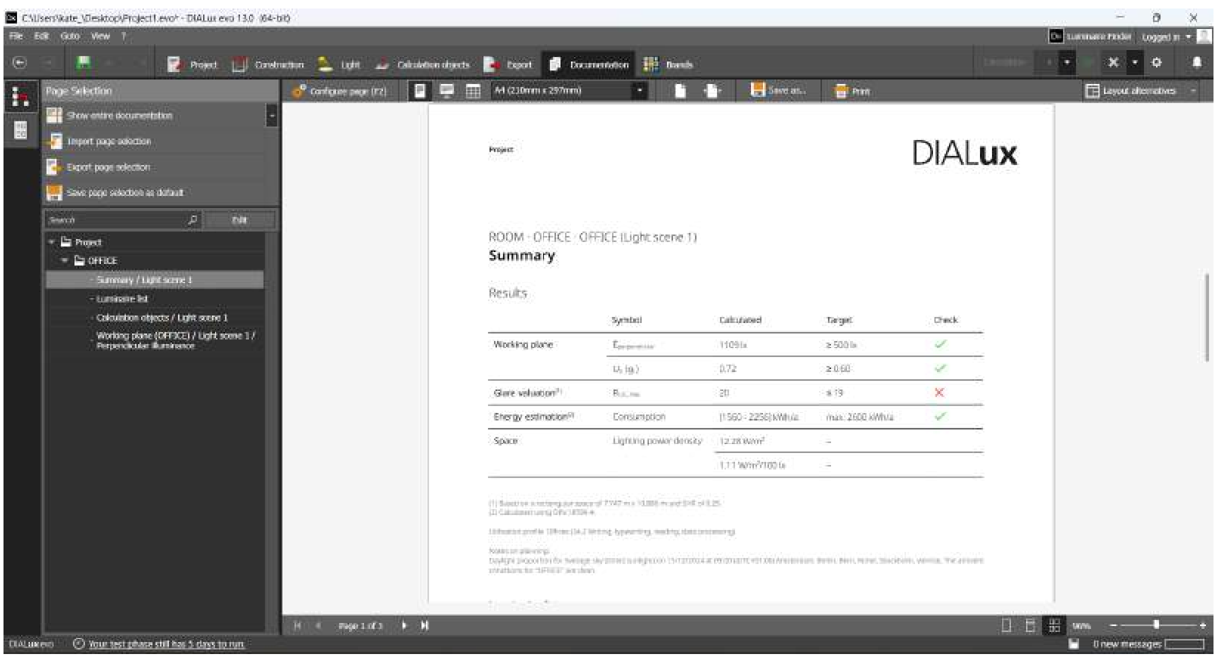

Figure 3: A snapshot from the 'Documentation' of DIALux, depicting the results for a random example.

company, and the specifications of interest are given in Table 2.
Another thing important to note is that there are calculating surfaces defined that cover each desk; see Figure 4. The scope of this is to calculate lux levels under the sensor and specifically on the desks, where we need to have indications above 500 lux, as determined by the EN 12464-1 rule.

Table 2: Philips lamp information

Specification	**Value**
Lighting Technology	LED
Luminous Flux	3,600 lm
Correlated Color Temperature	4000 K
Luminous Efficacy (rated)	123 lm/W
Optic type	Beam angle 90°
Unified glare rating CEN	19
Input Voltage	220-240 V
Power Consumption	29 W

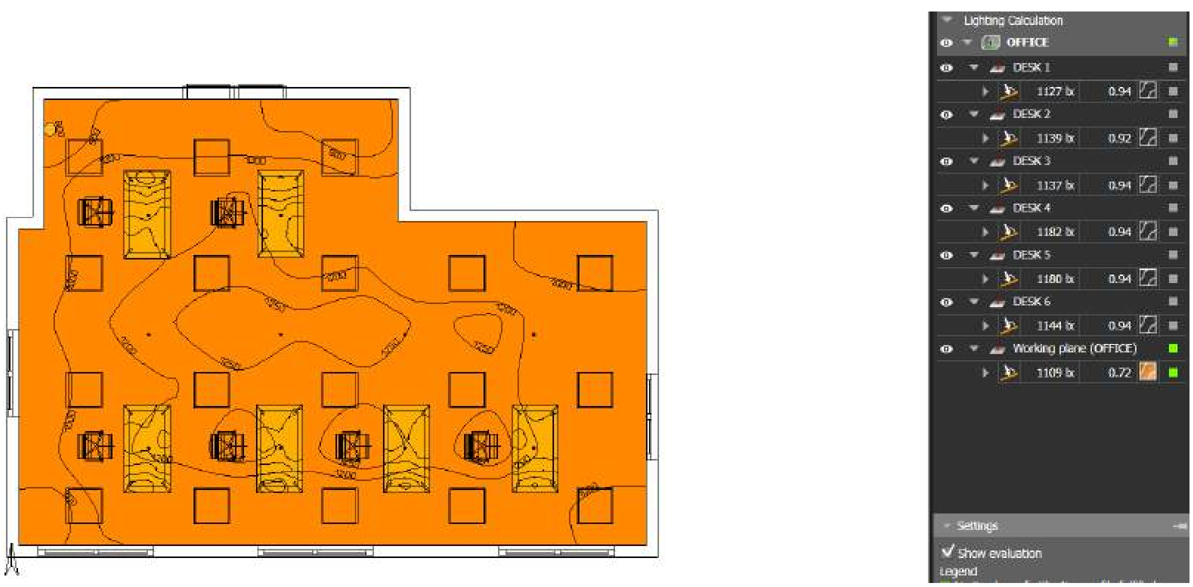

Figure 4: Desk surfaces are marked with the yellow color. On the left, information on the lux levels is given per calculating surface.

5 Experiments

This section presents the experimental setup used to evaluate the performance of the three selected optimization algorithms under varying sensor conditions. The objective is to assess how well each method balances energy efficiency and uniformity in a black-box, simulation-based smart lighting environment.

5.1 Examinations

For the experiments, we conducted a grid search over the three algorithms, two Bayesian Optimization variants and Particle Swarm Optimization, with selected hyperparameter combinations. Due to the long runtime of each optimization iteration, we limited the parameter configurations as shown in Table 3.

Table 3: Hyperparameter Grids for Optimization Algorithms

Algorithm	**Parameter**	**Values**
BO	n_initial_points	[5, 10]
	n_calls	[30, 50]
	acq_func	[EI, PI]
PSO	swarmsize	[3, 5]
	maxiter	[9]
	omega	[0.5, 0.7, 0.9]
SAMO-COBRA	initial_points	[5, 10]
	max_iterations	[30, 50]

From this table, we can calculate the total number of experimental runs per algorithm: BO with Scikit involves 8 runs, PSO is executed 6 times, and SAMO-COBRA is tested with 4 configurations, totaling an amount of 18 experiments. All algorithms are evaluated using an equivalent computational budget to ensure fair comparison. In the case of PSO, the total number of evaluations is computed as $s \times i + s$, where s is the swarm size and i is the number of iterations. For example, for a swarm size of 3, the total number of evaluations becomes $3 \times 9 + 3 = 30$.

Note that in Section 6, the naming convention used to describe the algorithm configurations follows a consistent format across all methods:

- **BO**: Configurations are denoted as `EV_IN_ACQ`, where `EV` is the number of evaluations, `IN` is the number of initial design points, and `ACQ` refers to the acquisition function used (e.g., EI or PI).
- **PSO**: Denoted as `SW_IT_OM`, where `SW` is the swarm size, `IT` is the number of max iterations, and `OM` corresponds to the omega (inertia weight) parameter.
- **SAMO-COBRA**: Represented as `EV_IN`, where `EV` again refers to the number of evaluations, and `IN` indicates the number of initial design points, following the same structure as BO.

This notation is used throughout the results section to concisely indicate the hyperparameter settings associated with each experimental run.

The experiments are executed across a diverse set of scenarios simulating different sensor availability states. The aim is to investigate the robustness of each algorithm when certain presence/motion sensors are disabled. The scenarios are:

- Scenario 1: All sensors are enabled
- Scenario 2–3: Two different randomized sensors are off
- Scenario 4–5: Four different randomized sensors are off
- Scenario 8–9: Six different randomized sensors are off
- Scenario 10–11: Eight different randomized sensors are off

In Figure 5, Scenarios 2 to 11 are visualized. Scenario 1 is excluded from the figure as it represents fully active and fully inactive settings. Red circles indicate disabled sensors, while active sensors remain unmarked.
Furthermore, for a scenario where PSO does not perform well and a BO variant does, we will rerun these BO variants without informative warm-start. This is conducted in order to observe the impact of a warm-start initialization and also make fair comparisons with PSO, as it is implemented without this feature.
Additionally, if an algorithm fails to find feasible solutions across most scenarios, we will consider refining its setup. This may include reducing the search space, increasing the evaluation budget, relaxing constraints, or modifying key hyperparameters. These decisions will be based on the outcome analysis.

5.2 Experimental setup

All experiments in this study were conducted on a personal laptop with the following specifications: an AMD Ryzen 7 4800H processor with Radeon Graphics, running at 2.90 GHz, and 16 GB of installed DDR4 RAM. The system operates on a 64-bit Windows environment with an x64-based processor architecture. While the DIALux simulations are time-intensive, this hardware setup was sufficient for managing the experimental workload. Our system is compatible with any machine capable of running DIALux evo; hence, any device is able to run this project.

5.3 Time per simulation iteration

The integration of optimized lighting configurations with DIALux is achieved through a UI-based automation pipeline, in which the system interacts programmatically with the DIALux interface to simulate and evaluate each configuration. This process involves the following steps: launching the DIALux project, selecting the luminaires present in the scene, modifying the luminaire parameters (lumens and wattage), performing lighting calculations, accessing the documentation view to preview the final results, and exporting the evaluation data and saving the updated project. The execution time may vary slightly due to specific conditions during the simulation process. In cases where the newly proposed lumen and watt values for a particular lamp remain unchanged from the previous iteration, the system bypasses the need to press the 'Apply' button in the DIALux interface, thereby saving a small amount of time.

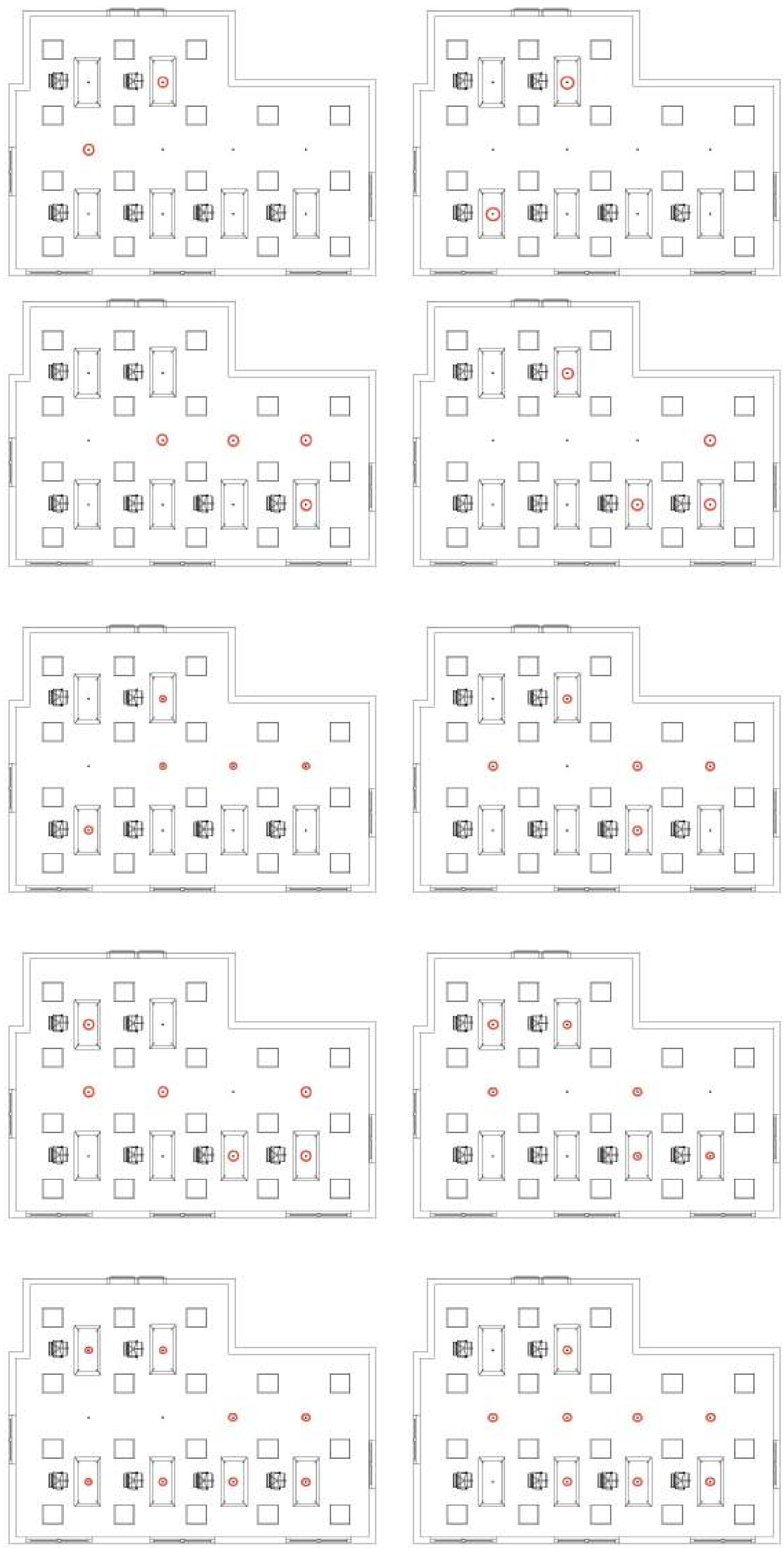

Figure 5: Overview of Scenarios 2–11. Red circles indicate disabled sensors.

Conversely, when changes are detected, this step is executed, contributing to longer iteration times.
Due to the UI-driven nature of this workflow, each iteration typically requires approximately 3 minutes to complete. However, it has been observed that, over time, DIALux encounters performance degradation, potentially due to internal caching, memory usage, or accumulation of process overhead. As a result, the duration of each iteration gradually increases, occasionally leading to significant slowdowns or even temporary stalling of the simulation environment. To address this issue, a monitoring mechanism was implemented to measure the duration of each iteration. If the time required exceeds a threshold of 3.5 minutes, the automation routine discards the current project file and initiates a new instance using a fresh project, thereby resetting DIALux's internal state. This practice restores the simulation speed to its baseline, maintaining consistent iteration times and ensuring the reliability of the optimization loop.

5.4 Reproducibility

To ensure the reproducibility of this work, all experiments were conducted using a fixed random seed of 42. This ensures that results can be consistently replicated under the same experimental settings. By controlling the random state across all algorithms and simulation runs, the impact of randomness is isolated, allowing for fair comparisons and reliable evaluations of algorithmic performance. The code can be found in the GitHub repository here.

The key performance metrics tracked include total energy consumption, desks illuminance, lighting uniformity, and penalty values related to lighting constraints. These results are discussed in detail in the following section.

6 Results

This section presents an analysis of the results obtained from the experiments described earlier. First, the outputs of any algorithm that encountered a feasible solution are summarized in both Tables and Figures, one for each scenario. These are followed by visualizations to further illustrate algorithm behavior and the impact of sensor data. As previously outlined, the evaluation focuses on three key metrics: energy consumption, desk illuminance (lux) over the six desk surfaces, and lighting uniformity.

6.1 Feasibility per scenario

Feasibility in this context is scenario-dependent, as it is influenced by the occupancy status defined for each case. While the penalty function remains consistent across all scenarios, we introduce an additional condition that prioritizes lighting adequacy for occupied desks. Specifically, a solution is considered feasible if all occupied desks receive at least 500 lux. Although the system also penalizes instances where vacant desks exceed this threshold, such violations are not grounds for infeasibility, and the solutions are still accepted. In essence, feasibility strictly requires compliance for occupied desks, whereas violations on unoccupied ones are penalized but tolerated accordingly. For instance, in the first scenario, which assumes full occupancy, every desk must reach a minimum of 500 lux for a solution to be considered feasible. In contrast, in the second scenario where the second desk is unoccupied, a lux value below 500 at that specific desk is acceptable, and the solution is still considered feasible.

Tables 4, 12-22 present the configurations executed for each algorithm that finds feasible solutions, along with their corresponding results in terms of energy consumption, lighting uniformity, and desk illuminance. In cases where a trade-off arises, such as one solution yielding higher uniformity but also higher energy consumption, and another offering the reverse, we include both. This approach ensures a balanced consideration of both objectives.

It is important to note that the presented results include two categories of solutions. The first category corresponds to the solution selected as optimal by BO, based on its objective function. In cases where this solution does not meet the feasibility criteria; either by failing to provide at least 500 lux on occupied desks or by falling below the acceptable uniformity threshold, we also report the first valid feasible solution identified with the lowest objective value. The second category includes the most representative and clearly superior feasible solutions per algorithm and hyperparameter configuration. Although additional feasible configurations were obtained during the optimization process, they are not included in the Tables if they exhibit both higher energy consumption and lower uniformity compared to other solutions derived under the same settings.

In cases where the outcomes from both budgets are identical or the solutions obtained with 30 evaluations are sufficiently satisfactory in terms of levels of uniformity and energy, the favored algorithm configuration is the one associated with the lower budget, thereby choosing the more computationally efficient one. Furthermore, solutions corresponding to the warm-start configuration are excluded, as the focus is on results discovered independently by the optimization algorithms rather than predefined inputs. Finally, there are scenarios where more than half desks are vacant; hence for those, we count solutions with uniformity below 0.6 as eligible and are still presented in the respective Tables.

Scenario 1

In the first scenario, PSO fails to identify any feasible solution, likely due to the absence of warm-start initialization. In contrast, BO and SAMO-COBRA successfully identify one or more feasible solutions, as summarized in Table 4.
Notably, several feasible solutions identified by Bayesian Optimization with a 30-evaluation budget also appear among the solutions discovered using the extended 50-evaluation budget. This observation suggests that the algorithm is often capable of converging to acceptable configurations within a relatively limited number of evaluations. However, increasing the budget may still yield additional or improved solutions beyond those found with fewer iterations.
Among the recorded results, the configuration using BO with 50 budget, 10 initial samples, and the EI acquisition function achieved the highest uniformity, with a value of **0.76**. The same configuration but with PI instead of EI, attained the lowest energy consumption, indicating a score of **427 W**. The lighting configuration corresponding to the highest uniformity was identified at iteration 49. In this solution, all lamps operate at full intensity (3600 lumens), except for lamp 11, which emits 664 lumens. This outcome is expected since the objective scopes to maximize uniformity without minding the increased energy consumption. Given that the scenario assumes full occupancy, it is reasonable that 17 out of 18 lamps are set to their maximum output in order to achieve an even distribution of light throughout the space. The slightly reduced output from lamp 11 may have been adjusted to correct for localized over-illumination or to improve balance across neighboring areas. In contrast, the lighting configurations for the result that brings the lowest energy were found in the 43rd out of 50 evaluations, and the rounded values are: [3600, 3131, 2888, 2735, 3474, 3539, 2140, 2347, 3600, 2597, 2540, 3600, 2772, 2678, 3482, 1150, 3124, 3600] for lamps 1-18 correspondingly. The variation in lumen values reflects a deliberate balance between meeting minimum lighting requirements and reducing energy usage where possible.
As the BO works in a Single-Objective manner, with a balanced sum of uniformity and energy, we also depict the optimal value achieved by all experiments of BO, that is **0.14**, with the lighting configurations being set to [3600, 3588, 2560, 3600, 2022, 2783, 3517, 896, 3519, 2201, 3551, 3600, 2629, 3193, 3600, 2115, 2926, 3600]. This solution is illustrated with a star (*) throughout the corresponding line in Table 4.

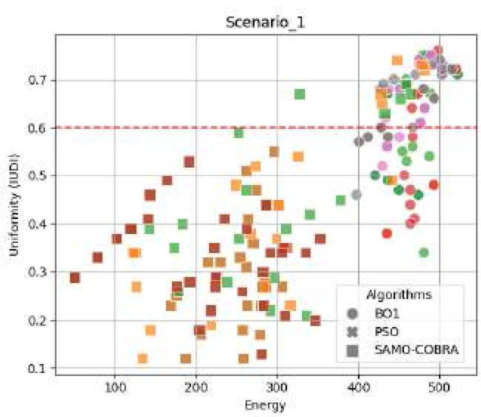

Figure 6: Energy vs Uniformity for Scenario 1

Table 4: Feasible solutions for Scenario 1

Algorithm	Settings	Energy	Uniformity	Desk Lux per surface
BO	30_10_EI	520.3	0.72	[648, 654, 652, 688, 683, 670]
		473.2	0.67	[632, 631, 536, 580, 600, 656]
	30_10_PI	500.4	0.74	[621, 636, 635, 661, 657, 633]
		480.3	0.68	[598, 607, 627, 654, 602, 597]
	30_5_EI	505.2	0.73	[645, 652, 643, 666, 628, 631]
		455.0	0.69	[625, 551, 642, 577, 504, 526]
		435.8	0.67	[595, 508, 628, 562, 518, 550]
	30_5_PI	489.6	0.75	[589, 614, 628, 648, 629, 633]
		474.9	0.74	[596, 599, 592, 611, 606, 630]
		460.7	0.69	[584, 578, 559, 579, 574, 623]
		450.5	0.68	[541, 564, 552, 582, 564, 617]
	50_10_EI	498.4	**0.76**	[642, 638, 649, 581, 586, 659]
		473.2	0.67	[632, 631, 536, 580, 600, 656]
		466.1	0.64	[636, 600, 594, 582, 587, 611]
	50_10_PI	430.9*	0.69*	[586, 616, 599, 527, 522, 519]*
		490.3	0.73	[612, 634, 615, 629, 612, 633]
		470.6	0.71	[602, 613, 613, 598, 612, 591]
		443.6	0.70	[560, 511, 617, 578, 589, 628]
		500.4	0.74	[621, 636, 635, 661, 657, 633]
		427.0	0.68	[576, 575, 562, 501, 518, 586]
	50_5_EI	479.6	0.75	[581, 567, 630, 639, 647, 623]
		458.6	0.70	[561, 594, 569, 610, 623, 597]
		455.0	0.69	[625, 551, 642, 577, 504, 526]
		435.8	0.67	[595, 508, 628, 562, 518, 550]
	50_5_PI	489.6	0.75	[589, 614, 628, 648, 629, 633]
		474.9	0.74	[596, 599, 592, 611, 606, 630]
		460.2	0.69	[565, 568, 561, 582, 617, 635]
		450.5	0.68	[541, 564, 552, 582, 564, 617]
		433.4	0.62	[510, 548, 584, 570, 518, 557]
SAMO-COBRA	50_10	458.1	0.69	[588, 615, 520, 568, 601, 600]
		451.7	0.66	[509, 574, 509, 595, 655, 638]
	50_5	447.2	0.74	[551, 571, 565, 590, 520, 546]

Scenarios 2-3

Scenarios 2 and 3 (presented in Tables 12 and 13, respectively) explore the case in which two sensors do not detect occupancy. In both scenarios, PSO is unable to identify any feasible solution, whereas BO and SAMO-COBRA successfully return such results.
A consistent pattern emerges across both scenarios: BO using the PI acquisition function reliably discovers feasible solutions, whereas configurations employing EI are less effective. Specifically, in Scenario 2, no feasible solutions were found using EI-based configurations. In Scenario 3, only one EI-based configuration yields a feasible result, and that one is identical across both the 30- and 50-evaluation budgets. Regarding SAMO-COBRA, feasibility in both scenarios is achieved exclusively under the 50-evaluation budget. The whole set of solutions is

presented on the first two plots in Figure 7.
In terms of lighting performance, SAMO-COBRA with 50 evaluations and 5 initial points produces the highest uniformity in both cases: **0.78** in Scenario 2 and **0.74** in Scenario 3. Notably, in both scenarios, the 50-5-PI setting of BO yields the lowest energy consumption, **437.6 W** and **392.6** respectively, for 2 and 3. In Tables 12 and 13 the best solutions identified by the BO process are indicated with a star. The optimal score, which is feasible, for the second scenario is **0.11**, while for the third we present both the optimal **0.48** and the first valid optimal **4.0**, which includes the lowest energy too.
The corresponding lighting settings for the highest uniformity are [3600, 3600, 3600, 2793, 3600, 3600, 3600, 3600, 3600, 3600, 2792, 2437, 2556, 3568, 2940, 2837, 3600, 3260] for Scenario 2 and [3600, 3600, 3600, 2774, 3600, 3600, 3600, 3600, 3600, 3600, 3600, 2595, 3600, 2958, 3600, 2947, 3600, 3092] for Scenario 3. Regarding energy, BO discovers the following lumens: Scenario 2 with [3600, 2552, 3540, 3600, 2365, 3545, 1892, 3084, 3600, 2304, 2591, 3240, 1078, 3538, 3182, 3600, 3380, 3600], Scenario 3 with [3600, 2300, 2244, 2795, 2598, 621, 2127, 3137, 3333, 2899, 2755, 3600, 3600, 3600, 2093, 2695, 2780, 1944]. The optimal value identified in Scenario 2 is given when lights are set to [3017, 2532, 3399, 3600, 2405, 1784, 3600, 3549, 2512, 1792, 2579, 3423, 3600, 3600, 2842, 3600, 3465, 3236], while the best valid for Scenario 3 is given when lights are set to [3600, 2432, 2118, 2556, 3598, 0, 2281, 3298, 832, 2497, 2833, 3600, 3600, 3600, 3177, 2683, 2884, 3575].

Scenarios 4-5

These two scenarios represent the case in which four sensors detect inactivity. Once again, PSO fails to identify any feasible solutions, whereas both BO and SAMO-COBRA demonstrate greater robustness under these conditions. A total of 14 feasible solutions were identified in Scenario 4, and 10 in Scenario 5.
Regarding Scenario 4 (Table 14), all BO configurations, except for the 30-10-EI setting, produced at least one feasible solution. In contrast, SAMO-COBRA succeeded in only one configuration, specifically with 50 evaluations and 10 initial points, suggesting difficulty in optimization and that maybe a higher budget is necessary for feasibility in the current scenario. If we view the BO process as a Multi-Objective, then we would choose the 30-5-PI setting, as it yields the highest uniformity (**0.72**) in less budget, and 50-10-EI for depicting the lowest energy (**428.4 W**), overcoming SAMO-COBRA for both objectives. However, as BO is Single-Objective, we mention the result that returns the lowest balanced sum between these two objectives and is **0.42**, and for the optimal valid is **1.58**. The settings for highest uniformity, lowest energy, and optimal valid solutions are: [3391, 3516, 3485, 3437, 3406, 3311, 3384, 3505, 3439, 3339, 3436, 3527, 3511, 3509, 3519, 3505, 3516, 3492], [3600, 2543, 3600, 3600, 3600, 1569, 1582, 2007, 3600, 3600, 0, 3600, 3288, 2586, 3600, 3600, 3600, 3600], [3600, 3047, 3600, 3600, 3600, 3600, 3600, 2851, 3600, 3600, 3600, 2138, 1902, 3600, 2944, 2467, 3600, 2801], respectively, showcasing really similar lighting configs.
Scenario 5, presented in Table 15, exhibits slightly different behavior. Here, BO showcases a single feasible configuration when using 5 initial points with the PI acquisition function, regardless of whether the evaluation budget was 30 or 50. Another solution is given by 50-5-EI hyperparameters, which happens to be the optimal valid one, i.e., that has the lowest objective value. The fourth result is not feasible, but is chosen as the best option from BO. SAMO-COBRA performs more effectively in this case, as it discovers five feasible solutions, of which four are discrete. It is also able to identify lower energy compared to what BO brings.

Considering BO as Multi-Objective, the selected solution for the highest uniformity is decided among three runs (two of each are identified by BO) that indicate **0.72** uniformity. However, both BO solutions (487.1 W) showcase higher energy than SAMO-COBRA (473.7 W) with the same level of uniformity; hence, we exclude the first two and choose the last one. Concerning energy, SAMO-COBRA with 50 evaluations and 5 initial samples identifies the lowest among all experiments, with a total wattage of **414.6**. Although, due to the Single-Objective setting of BO, the retrieved optimal value is **1.19**, and for the optimal valid is **17.32**. The high difference here lies in the given penalty for emitting over 500 lux on desks 2 and 5. The lighting configurations of SAMO-COBRA with highest uniformity are [3297, 3600, 3247, 2730, 2133, 3600, 3600, 3355, 3600, 2867, 3600, 2709, 3600, 3600, 3600, 3299, 3600, 2757] , and for lowest energy are [3600, 3600, 2768, 3021, 3600, 3600, 3274, 2784, 3600, 3600, 2583, 2177, 3600, 3600, 3600, 2100, 2575, 2376]. The best valid solution indicates lumen values of [3600, 3600, 3600, 3600, 3600, 3600, 3600, 3600, 2723, 3600, 3600, 0, 1424, 3600, 1673, 3600, 3600, 3600].

Scenarios 6-7

The examined scenarios reflect conditions where the half sensors report vacancy. Under such settings, PSO consistently failed to produce feasible outcomes. On the other hand, both BO and SAMO-COBRA proved to be more resilient, presenting 15 feasible solutions for each scenario.

In Scenario 6, summarized in Table 16, BO when using 10 initial points with the EI acquisition function does not identify any feasible configuration, regardless of whether the total evaluation budget was 30 or 50. SAMO-COBRA with 30 evaluations lacks finding feasibility. In contrast, bigger budget discovers some solutions. This indicates that BO can efficiently detect isolated feasible points early, whereas SAMO-COBRA gains more from a larger budget in these conditions.

SAMO-COBRA yields the optimal uniformity, with a score of **0.74** and configs [3600, 3600, 3600, 2774, 3600, 3600, 3600, 3600, 3600, 3600, 3600, 2595, 3600, 2958, 3600, 2947, 3600, 3092]. The lowest energy consumption is recorded by BO with 50 budget, 10 initial points and PI, and it is **409.2 W** with lights set to [3600, 2780, 625, 1074, 3600, 2419, 581, 876, 2127, 0, 2444, 3600, 3584, 1499, 3600, 3600, 3600, 2303]. This applies when we do not take into account the first valid solution with the lowest total objective value, which is **392.6 W**. The selected configuration by BO brings the second lowest energy but a uniformity below the predefined threshold of 0.6. Hence, the next best solution, with light settings [3600, 2300, 2244, 2795, 2598, 621, 2127, 3137, 3333, 2899, 2755, 3600, 3600, 3600, 2093, 2695, 2780, 1944], that depicts feasibility is also in the Table. The objective values for each of these cases are **0.49** and **4.0**.

The second scenario of this category, shown in Table 17, identifies more solutions for the BO with 5 initial points and PI settings, compared to the previous one. However, it is unable to find any results with EI as the acquisition function. SAMO-COBRA is again experiencing difficulty in recognizing plenty of solutions, with only two discrete outcomes. The first and third are the same, found with 5 initial points and a budget of 30 and 50, respectively. The second solution indicates lower energy, but also lower uniformity.

The most successful, in terms of efficiency, setting seems to be BO with 50 iterations, 5 initial points, and PI, as it is capable of locating both the lowest energy and highest uniformity, in case we take BO as a Multi-Objective algorithm. The numbers for energy and uniformity are

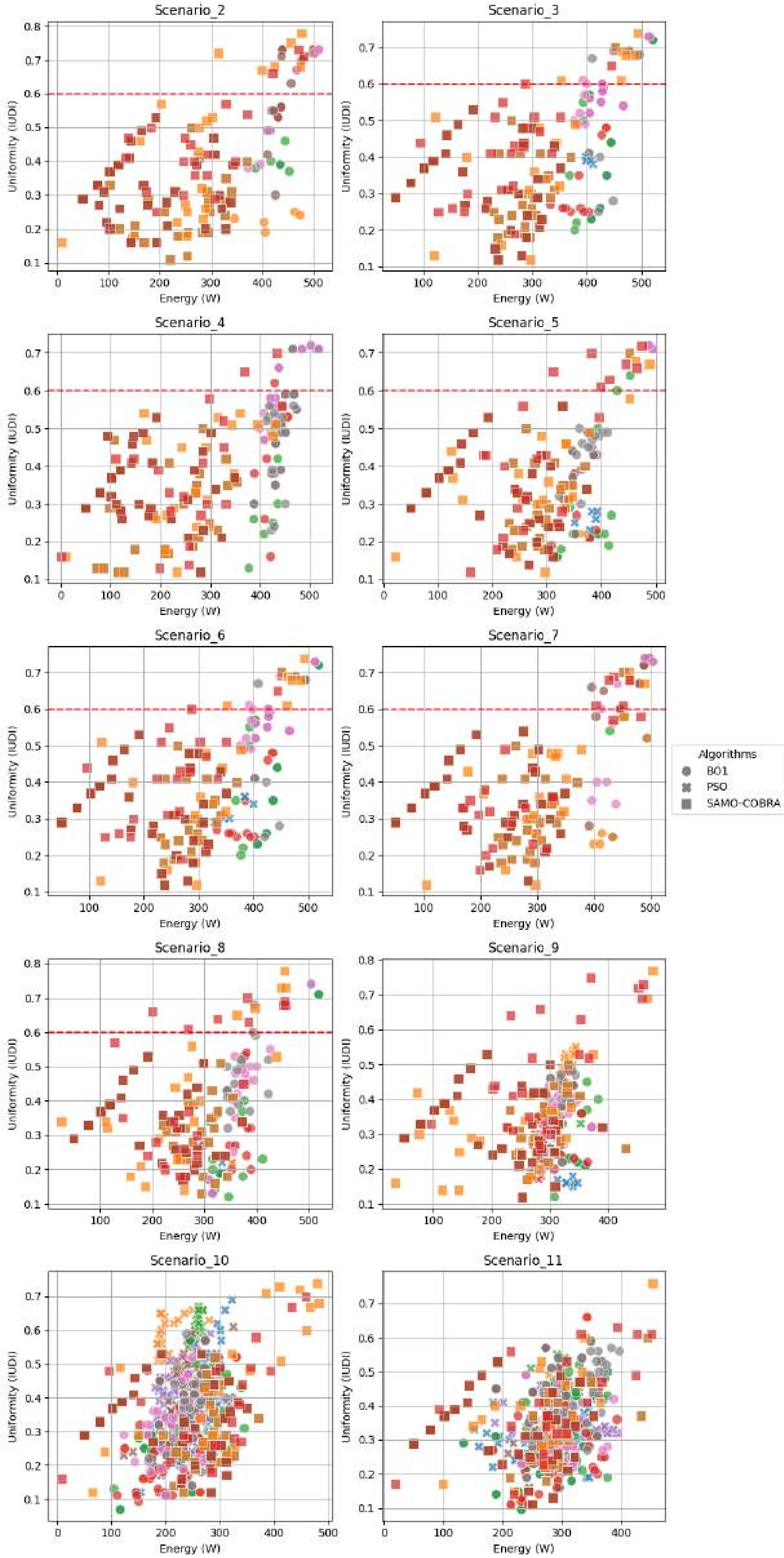

Figure 7: Energy vs Uniformity for Scenarios 2-11

416.5 W and **0.74**, respectively. The corresponding light settings are: [3600, 3383, 2625, 2415, 2295, 2945, 2766, 2651, 3600, 3600, 1672, 2675, 3600, 1671, 2600, 3600, 3052, 2900] and [3600, 3407, 3460, 2816, 2684, 3233, 3595, 3267, 3600, 2845, 3541, 3522, 3594, 3600, 3408, 3599, 3409, 3600]. The chosen best solution of BO, with an objective value at **0.79**, though, gives lower energy but infeasible uniformity; thus, we also present the first feasible solution that yields the lowest objective calculation, that is **3.0**, and depicts [3600, 3067, 3285, 1922, 2377, 3095, 1992, 2366, 3600, 3439, 1347, 2657, 3600, 1606, 2593, 3179, 2589, 2782] lumens for lamps 1-18.

Scenarios 8-9

Scenarios 8 and 9 are the first to have more than half (6) of the sensors depicting unoccupancy. Scenario 8 illustrates inactivity on three desks, while Scenario 9 has four desks vacant. For the eighth scenario, we will proceed as with the previous ones, but for the ninth scenario we introduce a new logic rule concerning uniformity. Meaning that, as more than half of the desks are vacant we do not care that much having uniformity above 0.6, thus we also accept solutions with lower uniformity scores.
Three discrete solutions are found by BO, with the setting of 10 initial points not working that well for the present scenario. SAMO-COBRA finds feasibility in only one solution under 50-evaluation budget. The optimal energy is found by BO with an indication of **396.5 W**, while the highest uniformity is again depicted from BO (**0.74**), with both having PI as the acquisition function. The optimal value is identified by 50-10-PI setting and is **0.08**, with lights set to [3417, 3600, 3600, 1515, 1764, 3600, 3600, 3600, 2009, 2822, 3581, 1197, 966, 3600, 3600, 2378, 1269, 3108]. This matches with where the vacant and occupant desks are located in the space, as the occupied depict higher emission of light, compared to those that are vacant.
In the case of Scenario 9, all the solutions presented in the scatter plot of Figure 7 might be acceptable based on how much uniformity the user of this system wants to apply in such conditions. In Table 19 we present only the optimal solutions returned by BO and PSO, as well as all the solutions identified by SAMO-COBRA. In this case, SAMO-COBRA may bring solutions that have higher energy as it tries to achieve the threshold defined before for uniformity. BO's chosen solution yields an optimal value of **1.14** with [2280, 1595, 3600, 2914, 3593, 643, 2152, 2072, 3561, 3600, 3090, 461, 2909, 2148, 3138, 1938, 432, 1402] as lumen values. On the other hand, PSO optimal solution exports a score of -**0.08** with lamps set to [2310, 1404, 3002, 2417, 1932, 1572, 575, 2340, 2831, 1052, 5, 2059, 2242, 1426, 1592, 1537, 1918, 3450].

Scenarios 10-11

These two scenarios indicate a condition where 8 out of 10 sensors are disabled. It is obviously difficult to adjust the lighting when using the same penalties and objectives for this problem, as most of the desks, if not all, are not occupied.
Scenario 10 is a special one, where no desk is occupied, but two sensors in the corridor identify human activity. This means that we do not have to account for desks with a minimum lux value of 500 or have uniformity above 0.6. The solutions here vary, but probably for this case we only want the lowest energy, which is found by the SAMO-COBRA 50-10 setting, and it is **8.0 W**, with uniformity at 0.16. For this setting, the lighting configurations are all near to zero. The next solution with the lowest energy, which is identified by three SAMO-COBRA

variants, is the one that yields energy at 49.6 W; indicating uniformity of 0.29. This solution might be more suitable as there is still occupancy detected, and the configurations of it are: [1800, 116, 97, 87, 83, 76, 67, 61, 59, 1200, 720, 514, 327, 276, 211, 189, 156, 124].
The final scenario depicts occupancy on two desks, hence we care about the uniformity level and the rule of 500 lux on these two specific desks, in contrast to the previous scenario. Therefore, the optimal solution found by BO is not valid, as the first and third desks are not illuminated properly. PSO again struggles to find any solution. SAMO-COBRA though, discovers some valid and compatible with our needs solutions. The lowest energy is identified by SAMO-COBRA's variant of the 30-5 setting, being **317.1 W** and the corresponding uniformity is 0.37. In case the users of this system want to choose higher uniformity levels, they should compromise with an increase of 110 W of energy usage, unless the solution by BO is preferred.

6.2 Convergence over iterations

Although the optimization process is not inherently designed to improve monotonically over time, we visualize the behavior of each algorithm across iterations, regardless of whether feasible solutions were ultimately found. For each scenario, we present two convergence plots: one corresponding to a 30-evaluation budget and one to 50 evaluations.
In these plots provided in Appendices, it can be observed that configurations initialized with 5 initial points share the same starting evaluations across both budget settings. Similarly, configurations using 10 initial samples replicate the first five points, followed by five additional consistent points. We present the plots for the first scenario (Figures 8 and 9) as an example. The presence of the warm-start is also clearly visible as the first iteration in all relevant plots. In the energy plots, the majority of configurations achieve lower energy consumption compared to the warm-start, as evidenced by the downward trend of the curve below the initial value, an indication of improvement toward the optimization goal. For uniformity, the objective is for the curve to surpass the warm-start level. Even in cases where this does not occur, the initial uniformity values exceed 0.7, suggesting a relatively high baseline performance from the outset. It is important to note that for PSO, the first iteration shown corresponds to the warm-start configuration. However, this point is included solely for comparative purposes. PSO does not incorporate warm-starting in the same way as Bayesian Optimization methods; rather, it uses this configuration as a fixed reference point before beginning its standard evaluation process.

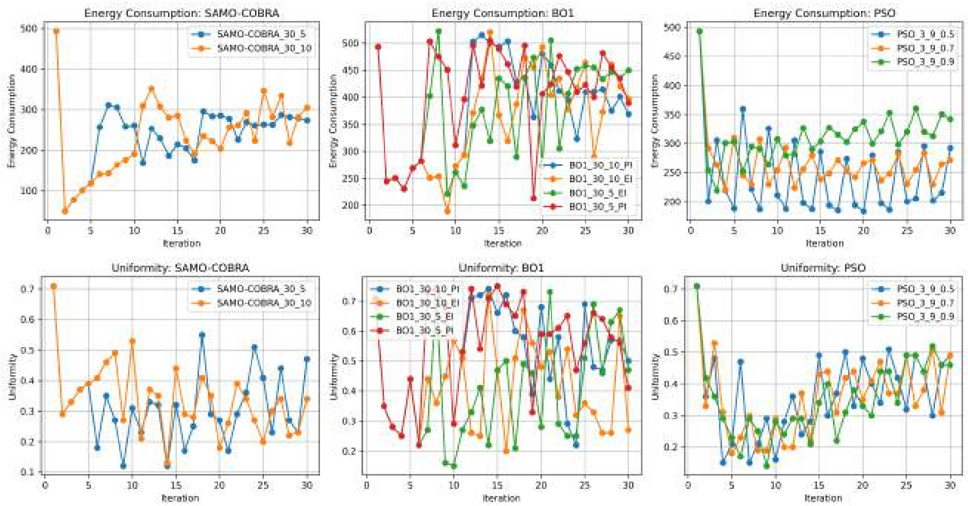

Figure 8: Energy and uniformity for Scenario 1 with 30-budget

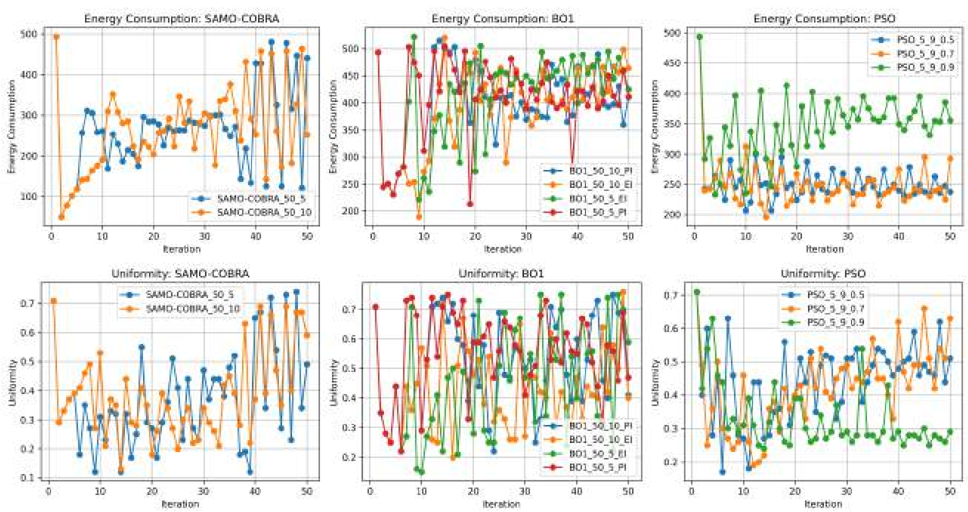

Figure 9: Energy and uniformity for Scenario 1 with 50-budget

6.3 Convergence of function evaluations (Single-Objective)

For this type of plots, we utilize the IOHanalyzer platform [9], and specifically the Expected Target Value (ETV) plots. These provide a comparative view of how efficiently and consistently each algorithm improves the objective value over time. From these plots, we can observe the convergence trends across multiple runs, offering insights not just into the speed of improvement, but also the robustness of each method. Algorithms with curves that descend more steeply early on demonstrate faster initial progress, while those that continue to improve steadily reflect stronger long-term performance. Overall, the ETV plots help highlight the trade-offs between convergence speed, consistency, and final solution quality across the evaluated methods. It is essential to mention that the solutions may not be feasible, hence the optimal value is what the algorithms chose with no other criteria added, which were explained before.

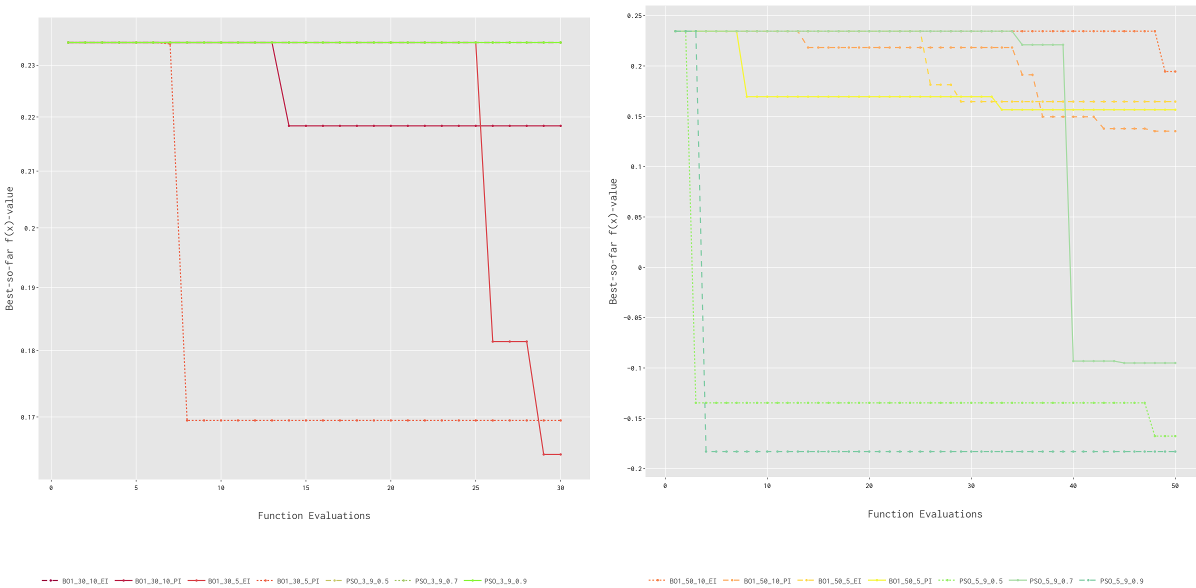

Figure 10: Function evaluations of Scenario 1 for both 30- and 50- budget

Scenario 1

Scenario 1 function evaluations are provided in Figure 10. BO-30-5-PI is the earliest to reach a promising solution, achieving an objective value of **0.17** quickly within the evaluation budget. BO-30-10-PI, which allocates more evaluations to initial exploration, improves slightly later, an expected outcome given its broader initial sampling. The most notable behavior, however, is observed in BO-30-5-EI, which struggles to make progress for the majority of the run. Only near the end does it make two sharp improvements, ultimately achieving the lowest objective value (**0.16**) among all algorithm variants. This late-stage convergence can be attributed to the characteristics of the EI acquisition function, which favors a more exploratory strategy early on, followed by exploitation once a promising region is identified. This behavior highlights EI's strength in finding high-quality solutions, albeit with a delayed payoff.
In the 50-evaluation budget setting, the three PSO variants yield negative $f(x)$ values, which are not considered valid based on the criteria discussed in an earlier subsection. BO, in contrast, demonstrates strong and consistent performance across all configurations. As observed previously, the distinction between PI and EI acquisition functions is evident with PI-based configurations tending to identify better solutions earlier. Specifically, BO-50-5-PI exhibits the earliest convergence and continues to improve until the end of the budget, ultimately reaching an objective value of **0.16**. Its counterpart with 10 initial points, BO-50-10-PI, shows a sharp improvement in the final evaluation phase, achieving the best result with a value of **0.14**. Meanwhile, BO-50-10-EI displays delayed progress, with gains occurring only in the late stages, leaving insufficient time to further exploit promising regions. BO-50-5-EI performs reasonably well but does not surpass its PI-based counterparts. Notably, in this scenario, EI-based configurations fall short of the performance achieved by PI-based ones, suggesting that PI's exploitation-focused strategy was more effective within the given budget.

Scenarios 2-3

The function evaluations for Scenarios 2-3 are illustrated in Figures 11 and 12, respectively.

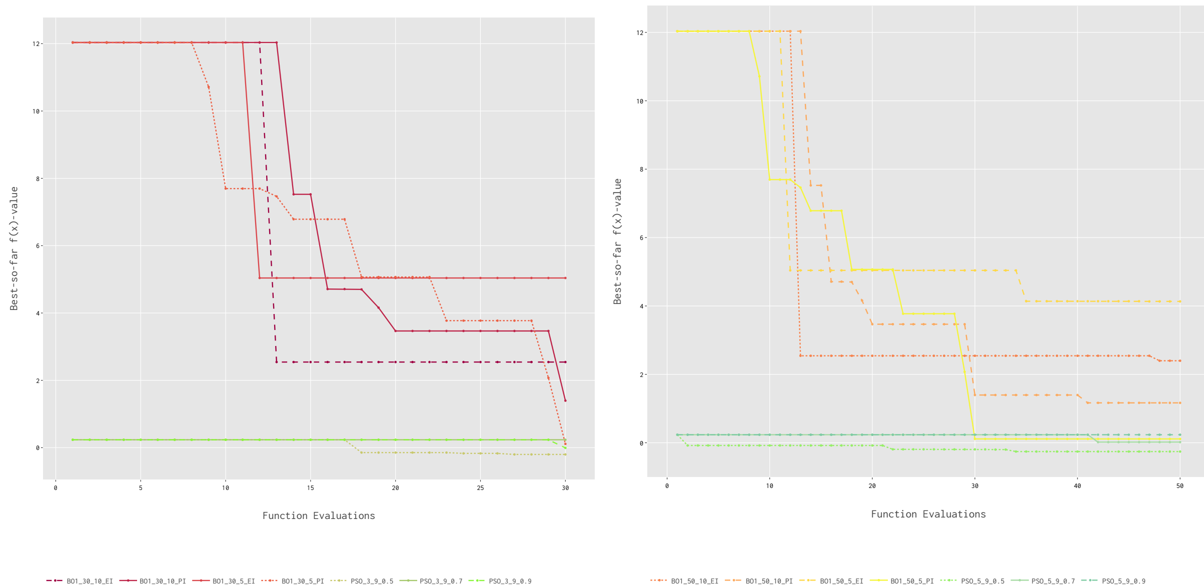

Figure 11: Function evaluations of Scenario 2 for both 30- and 50- budget

In Scenario 2 with 30 evaluations, the objective value calculation of BO starts from high (12) and all variants' curves decrease, some more and some less. It is clear that the one that improves the earliest also gives us the optimal value, which is **0.11**. This configuration is provided by BO-30-5-PI. The plot shows that in this case, PI does not always demonstrate the earliest convergence, as BO-30-5-PI is the last to optimize. Regarding 50 evaluations, the trends are very similar to those of the 30-budget until the halfway point of evaluations, not altering much after that point. The same solution is identified as best for this budget, meaning that BO-50-5-PI is the best-performing algorithm for the current scenario.
The next scenario, presents again similar trends concerning the 30- and 50- budget. As before, PI with 5 initial points is able to minimize the objective value earlier than others. However, in the first plot, BO-30-10-PI slightly surpasses the previous configuration, exhibiting **2.94** score, while in the second plot, the first configuration mentioned achieves a better score (**0.49**) than all, but it is very close with the 50-5-EI setting, having an indication of 0.6.

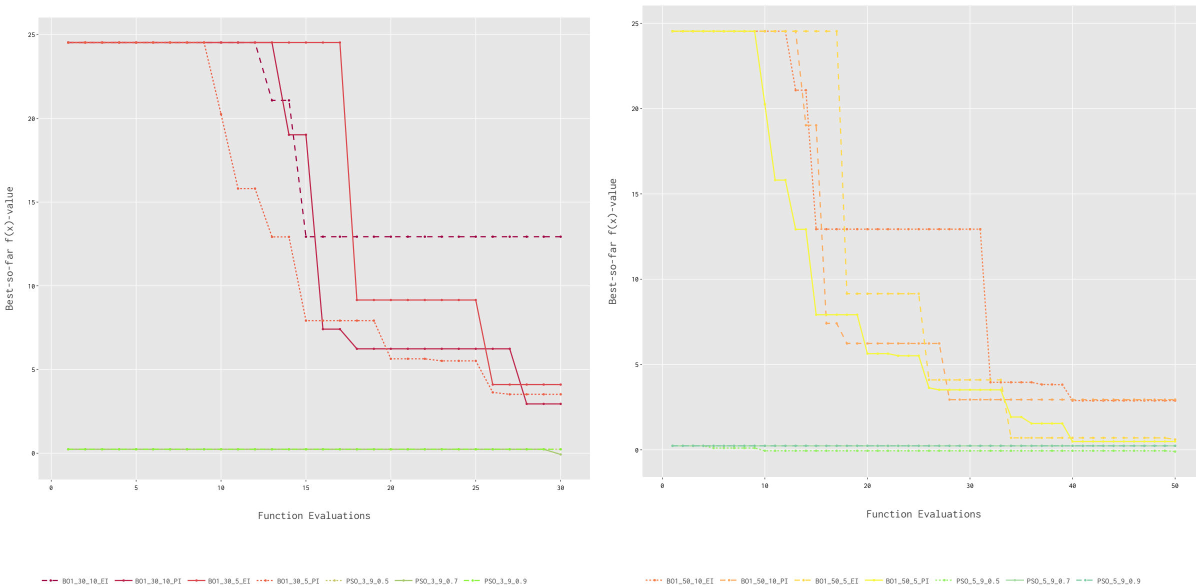

Figure 12: Function evaluations of Scenario 3 for both 30- and 50- budget

Scenarios 4-11

The following scenarios continue with the same logic, thus the plots will be provided in Appendices, without explaining further. In cases where the plots are really close to each other, we also show the specific optimal value of each algorithm.

6.4 Pareto Front (Multi-Objective)

In this section, we present the Pareto fronts of SAMO-COBRA generated for all scenarios. In this context, we treat energy consumption and uniformity as separate objectives rather than as components of a combined or weighted objective function. Here, 30 and 50 evaluations are not separated as before, thus there will be curves that might stop earlier. This might happen as a bigger budget may be able to find better solutions.

Scenario 1

In Scenario 1, we can validate what we have already observed in Section 6.1. Highest uniformity is represented by 50-5, exceeding 0.7, with energy around 450 W. The lowest energy comes from all variants, with wattage levels below 50. If we observe the lowest energy out of feasible solutions, we then should look at the second point from the top, again found by the 50-5 setting. It is also clearly visible that only the 50 evaluations are surpassing the desired uniformity for this scenario.

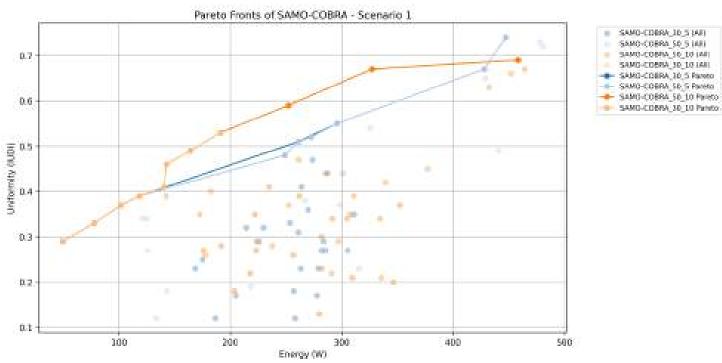

Figure 13: Pareto front for Scenario 1

Scenarios 2-3

Both scenarios are demonstrated in Figure 14. In the following scenario, the 50-5 configuration once again achieves the highest uniformity, approaching a value of 0.8. Among the solutions that meet the minimum uniformity threshold, the lowest energy consumption is approximately 310 W, also achieved by the 50-5 configuration. However, the 50-10 setting demonstrates comparable uniformity performance while consuming slightly less energy, suggesting a competitive trade-off. Notably, only these two configurations produce feasible solutions under the current scenario.

In Scenario 3, the 50-5 configuration again delivers the highest uniformity, this time exceeding 0.7. Within the subset of valid solutions (i.e., those satisfying the uniformity criterion), the lowest energy value is achieved by the 50-10 configuration at approximately 290 W, followed by the 50-5 configuration with around 350 W. Unlike the previous two scenarios, the 30-5 configuration also meets the desired uniformity level in this case. However, it is possible that this solution fails to meet the desk-level illuminance requirements, potentially rendering it infeasible despite its promising uniformity score.

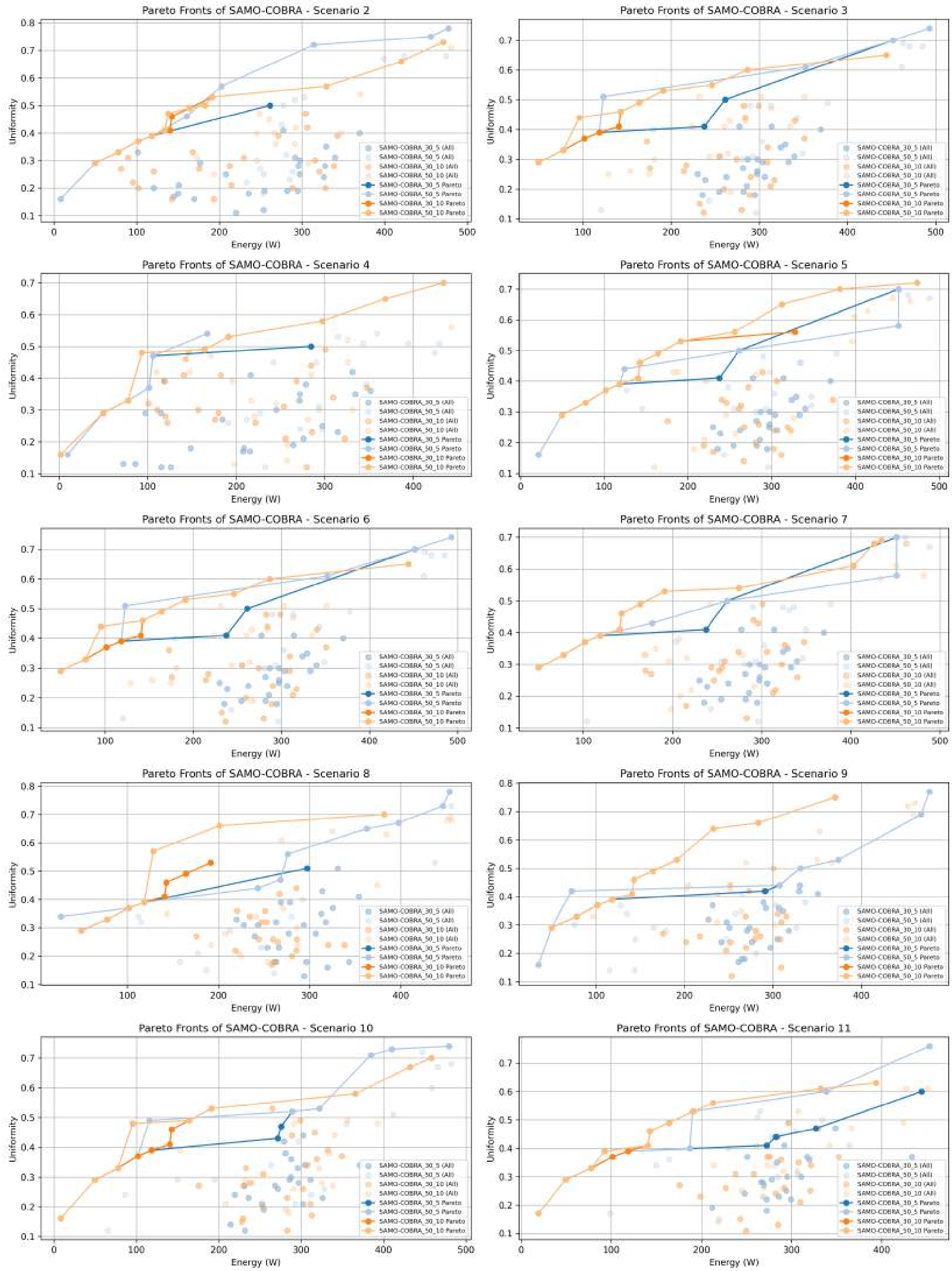

Figure 14: Pareto fronts for Scenarios 2-11

Scenarios 4-11

Scenarios 4 and 5 present 50-10 as the best algorithm regarding the uniformity, while for Scenario 4 it also appears to be the one with the lowest energy. For Scenario 5, 50-5 setting is the one showcasing the lowest energy. For feasible solutions, the lowest energy is given by 30-5 (and 50-5, as it is the same solution), reaching approximately 450 W.
Scenarios 6 and 7 show the 50-5 variant as the one with the highest uniformity and the 50-10 setting with the minimum energy. In general, the trends of these two are very similar, with the difference that the 30-5 setting in Scenario 7 demonstrates a solution with promising uniformity, achieving the same as 50-5 in just 30 evaluations.
The next Scenarios 8 and 9, showcase 50-5 parameters to succeed in recognizing both the minimum energy and maximum uniformity. In the eighth scenario, we care about having uniformity above 0.6, where only 50-5 and 50-10 locate solutions. The ninth scenario is kind of different from what we saw until now, as more than half of the desks are vacant; hence, there might be a slight preference for the solutions with lower uniformity, as they also give less energy.
Scenarios 10 and 11 conditions look alike with the previous scenario. Therefore, we may favor again the solutions with the least energy, even if they recognize lower levels of uniformity than the optimal one located. The lowest energy is demonstrated by the 50-10 configuration, with nearly zero wattage emitted for Scenario 10 and approximately 20 for Scenario 11.

6.5 Distribution of solutions

Since the distributed lumen values across lamps will serve as a reference for optimizing PSO configurations, we selectively present results for three representative scenarios: Scenario 1, where all sensors detect occupancy; Scenario 4, which captures the case where four sensors do not detect occupancy; and Scenario 8, where six sensors are inactive.
The boxplots in Figure 15 show how the lumens are distributed per algorithm and hyperparameter. We can also observe the differences in behavior for each run, as PSO, especially in Scenarios 1 and 4, depicts an entirely different pattern compared to BO. BO mostly explores in higher regions of lumen levels and drops throughout the scenarios as expected. SAMO-COBRA investigates larger regions, particularly in Scenarios 1 and 8, while for Scenario 4, the lumen values start low and reach up to mostly 2000 and 3000 lumens. PSO, as discussed previously, struggles to find solutions; thus, we need to use these diagrams as informative to improve its performance. We mostly get advice from BO instead of SAMO-COBRA, as this has a Single-Objective formulation.
In Figure 16, the bars are almost alike. Scenario by scenario, the zero lumen values increase as we expect, since the sensors detecting occupancy get less. Concerning the lumen value of 3600, it is understandable why in the first scenario there are approximately 3500 appearances, while we can see that this pattern is not observed for the other two. This might be interpreted by the fact that in Scenario 4 there are more lamps close to each other that are influenced by two or more sensors (as seen in Figure 17), compared to Scenario 8. Moreover, in Scenario 4 there are squares of lamps where the corresponding sensors are inactive, as this may cause lower lumen values than 3600. In addition, most lamps are located in the corridor in Scenario 4, compared to Scenario 8, where the lamps are also among the desks. Therefore, in order for the optimization process to identify solutions with valid desk lux illuminance, Scenario 8 needs more light illumination to satisfy this constraint, which is trivial for Scenario 4.

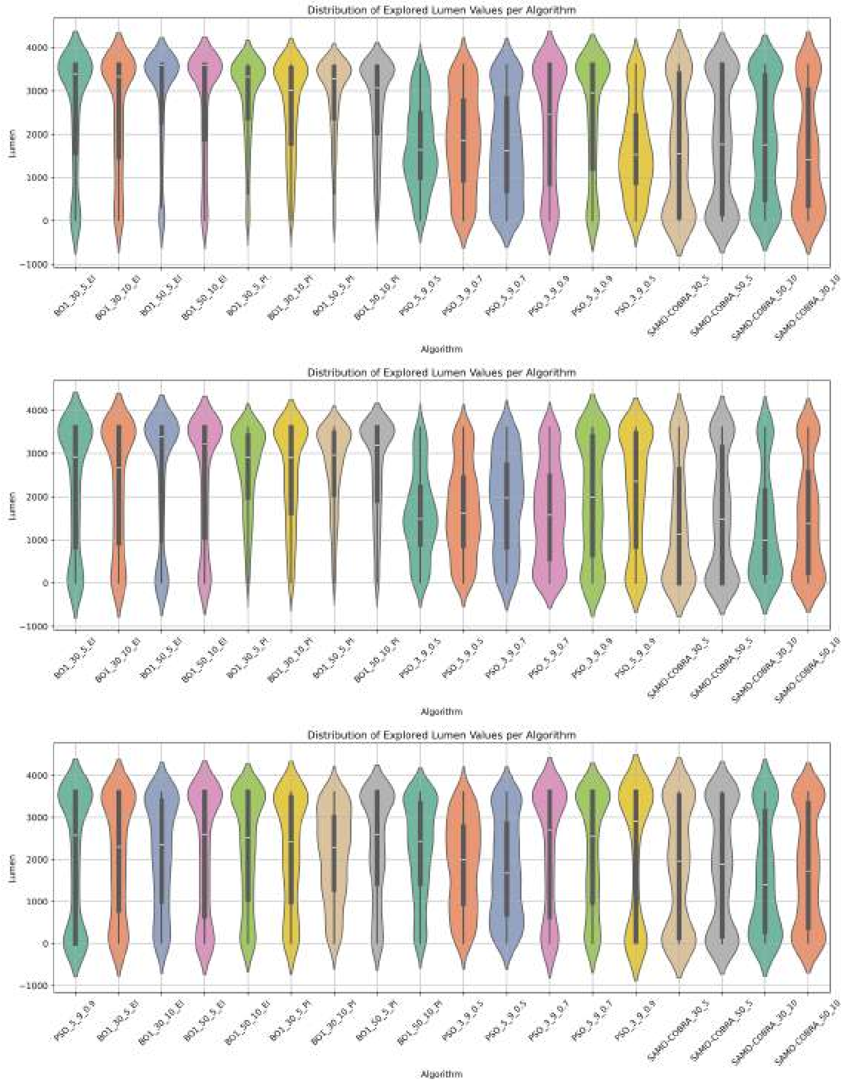

Figure 15: Explored Lumens per algorithm for Scenarios 1, 4 and 8

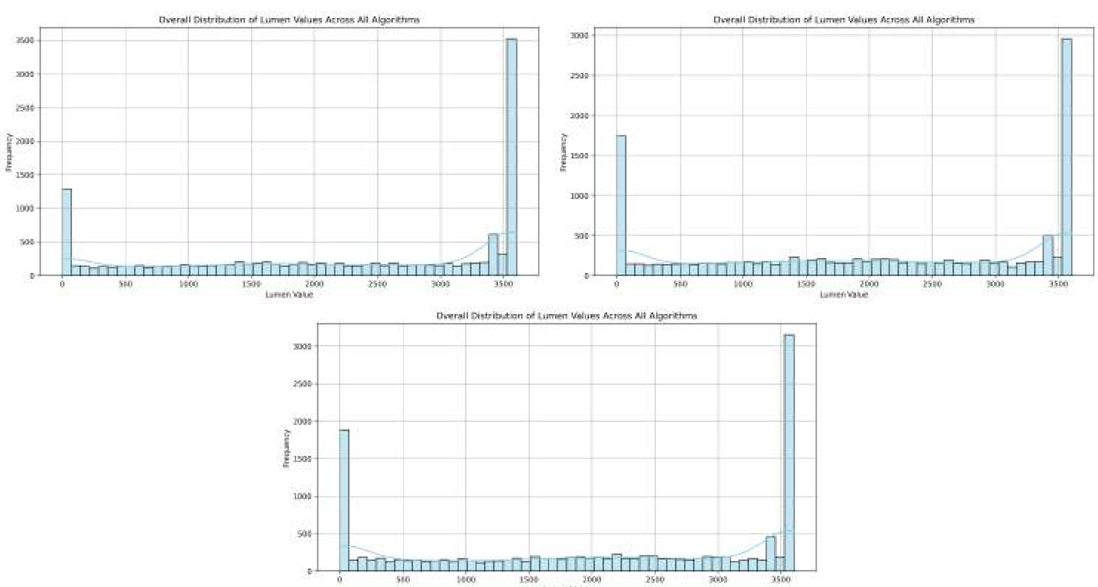

Figure 16: Explored Lumens over all algorithms for Scenarios 1, 4 and 8

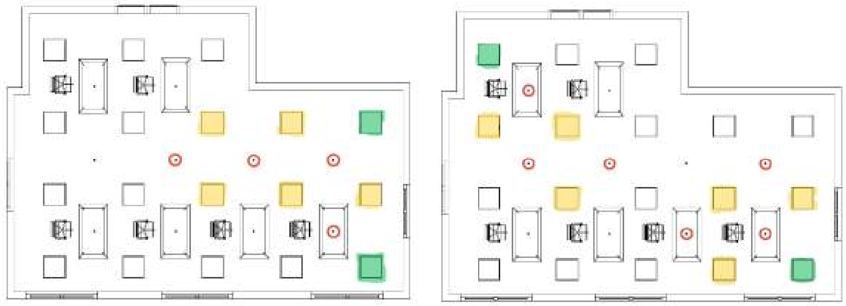

Figure 17: Showing the lights influenced by sensors in order to observe patterns with lumen values and conditions

After considering both Figures (16 and 15) we decided: For the first scenario, we minimize the range of the search space from (0, 3600) to (2500, 3600), since the most lumen values are distributed in this region. For the fourth scenario, we expand the search space a little due to the necessity of lower illuminance in some spots, and it is set to (1000, 3600). The last examined scenario concerns a state with more than half of the sensors not indicating occupancy. This means that low values are important for achieving the minimum possible energy. For this case, we will try to relax the constraint of uniformity, choosing uniformity above 0.40 instead of 0.60.

Enhancing PSO

Among the results of the enhancing of PSO, we will also show the runs of BO and SAMO-COBRA without a warm-start set. Actually, for BO the warm-start is entirely removed, while for SAMO-COBRA a dummy warm-start of 1800 lumens is given, as it was required to be provided. This procedure is done in order to observe whether BO and SAMO-COBRA are striving to find solutions as PSO, making them comparable. The results of both runs are given in Figure 18.

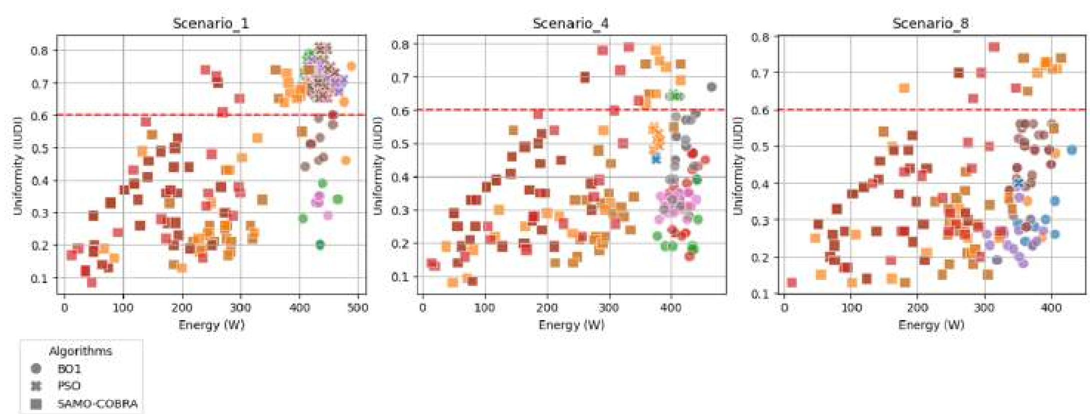

Figure 18: the allocation of solutions using the most recent iterations of BO, SAMO-COBRA, and PSO for Scenarios 1, 4, and 8.

Even though some of the illustrated solutions may not be feasible, this is how energy vs. uniformity plots are being formed. For Scenario 1, see Table 5, optimized PSO is now capable of identifying feasible solutions compared to the baseline PSO. It even exceeds the performance of BO, achieving a better optimal value of the Single-Objective function. The scores are **0.008** for enhanced PSO and **0.11** for warm-started BO, while BO with no warm-start achieves an indication of **0.18**, and finds a lot fewer solutions than the previous version of BO. SAMO-COBRA is unable to discover feasible solutions in these conditions, making it obvious that a warm-start is indeed essential.

For Scenario 4, demonstrated in Table 6, something interesting is happening. Both BO and PSO recognize solutions with very similar optimal values. These are **0.122** and **0.123** for PSO and BO, respectively. Compared to the previous version of PSO, PSO now finds at least two feasible solutions, while baseline PSO did not succeed at all. Regarding BO, it again struggles compared to the warm-started one but still succeeds in discovering feasibility.

For the last Scenario, provided in Table 7, where we tried to change the approach of optimized PSO, there is no feasibility observed. Additionally, no warm-started BO also fails. Although SAMO-COBRA encounters difficulty, it identifies a feasible solution. In general, this scenario is hard to examine, as we are not sure how much uniformity is necessary for half desks being occupied; hence, relaxing the constraint by setting the threshold of uniformity to 0.40 still did not help a lot.

Table 5: Feasible solutions for Scenario 1, where BO is set with no warm-start and PSO's search space is minimized.

Algorithm	Settings	Energy	Uniformity	Desk Lux per surface
BO	50_5_EI	488.6*	0.75*	[635, 641, 584, 590, 646, 626]*
		457.8	0.68	[587, 628, 616, 597, 528, 566]
PSO	3_9_0.5	428.3	0.71	[558, 531, 501, 562, 570, 574]
		428.6	0.72	[558, 530, 502, 563, 571, 575]
		428.9*	0.74*	[558, 530, 505, 563, 570, 575]*
	5_9_0.5	412.9	0.71	[531, 520, 581, 556, 505, 505]
		416.1	0.73	[534, 524, 586, 558, 508, 510]
		420.7	0.74	[540, 529, 594, 564, 511, 512]
		421.2	0.75	[539, 527, 583, 558, 524, 523]
		422.1*	0.76*	[541, 529, 577, 553, 528, 528]*
	3_9_0.7	406.9	0.73	[505, 501, 549, 542, 538, 534]
		411.2	0.74	[513, 505, 547, 552, 547, 546]
		411.9	0.77	[516, 512, 553, 540, 536, 537]
		415.1	0.78	[531, 520, 555, 541, 527, 535]
		416.7*	**0.79***	**[535, 522, 556, 542, 528, 535]***
	5_9_0.7	416.7	0.71	[518, 513, 525, 564, 546, 539]
		418.3	0.73	[518, 519, 522, 565, 550, 545]
		421.3	0.74	[524, 528, 520, 569, 554, 553]
		424.1*	0.75*	[530, 538, 520, 570, 557, 559]*
	3_9_0.9	414.4	0.74	[511, 554, 529, 534, 511, 526]
		414.6	0.75	[511, 554, 517, 532, 517, 534]
		421.0*	0.77*	[522, 572, 513, 529, 520, 549]*
	5_9_0.9	412.7	0.67	[557, 502, 550, 558, 530, 508]
		417	0.69	[578, 523, 515, 545, 541, 542]
		427.2	0.71	[564, 517, 540, 552, 574, 564]
		432.5*	0.81*	[546, 550, 520, 545, 575, 573]*

Table 6: Feasible solutions for Scenario 4, where BO is set with no warm-start and PSO's search space is minimized.

Algorithm	Settings	Energy	Uniformity	Desk Lux per surface
BO	30_10_PI	465.7	0.67	[605, 597, 590, 586, 624, 595]
	50_10_PI	465.7	0.67	[605, 597, 590, 586, 624, 595]
		398.6*	0.64*	[575, 555, 513, 539, 517, 421]*
PSO	3_9_0.9	**403.4***	**0.65***	**[522, 521, 541, 522, 505, 476]***
		402.7	0.64	[522, 524, 541, 524, 508, 466]

Table 7: Feasible solutions for Scenario 8, with SAMO-COBRA having as a warm-start a dummy solution

Algorithm	Settings	Energy	Uniformity	Desk Lux per surface
SAMO-COBRA	50_5	393.9	0.73	[520, 502, 514, 534, 509, 507]

6.6 Overview of algorithms

This section highlights the main insights of the algorithms in a Single-Objective and Multi-Objective manner. For the Single-Objective ones, we present the top-3 ranking with the corresponding mean and standard deviation, while for the Multi-Objective ones, we provide the Hypervolume and IGD+ metrics. Finally, we also compare the elapsed times of execution of each of the algorithms.

Figure 19 presents the Empirical Cumulative Distribution Function (ECDF) plots of the Expected Attainment Function (EAF) for the evaluated optimization algorithms. The x-axis represents the number of function evaluations, while the y-axis indicates the fraction of problem instances (or runs) for which an algorithm has attained at least a specified solution quality by that evaluation count. These plots effectively illustrate the speed and consistency with which each algorithm identifies high-quality solutions across multiple runs and scenarios.

Among the configurations tested under the 30-evaluation budget, the 30-10-PI variant demonstrates superior convergence behavior, outperforming all other algorithmic variants. Notably, 30-10-PI surpasses 30-5-PI, indicating that Probability of Improvement (PI) benefits from a larger initial sampling size in constrained budgets. Conversely, the Expected Improvement (EI) variants display the opposite trend: 30-10-EI underperforms relative to 30-5-EI, suggesting that EI may be less effective with increased initial sampling in low-budget cases. Overall, PI-based acquisition functions consistently outperform EI counterparts in the 30-evaluation setting, underscoring PI's robustness for time-limited optimization.

Regarding the 50-evaluation budget, during the early phase (10–25 evaluations), the 50-5-PI configuration exhibits superior performance, suggesting that fewer initial points facilitate faster early convergence. In the mid-phase (25–35 evaluations), 50-10-PI temporarily overtakes 50-5-PI, reflecting the dynamic trade-off between exploration and exploitation. However, by the final phase, 50-5-PI regains dominance, ultimately achieving the best solution quality. For EI, the 50-5-EI variant surpasses 50-10-EI after 25 evaluations, reversing their initial rankings. This indicates that EI's sensitivity to initial sample size diminishes as the evaluation budget increases.

Some general observations are that the EI variants with 5 initial samples are the slowest to reach the 0.9 attainment fraction threshold in both budget settings, while the PI-based configurations, again with 5 initial samples, demonstrate the fastest attainment, highlighting PI's efficiency regardless of evaluation budget. Furthermore, PI acquisition functions dominate performance across both evaluation budgets, while EI variants exhibit volatile rankings dependent on the initial sample size, performing suboptimally with larger initial samples under limited budgets but improving as the budget increases. Finally, all PSO variants show no progression beyond the warm-start solution within both evaluation limits, indicating ineffective adaptation under strict evaluation constraints or a small budget. This suggests PSO is unsuitable for problems with tight evaluation budgets and that EI should be reconsidered only when evaluation budgets exceed 50.

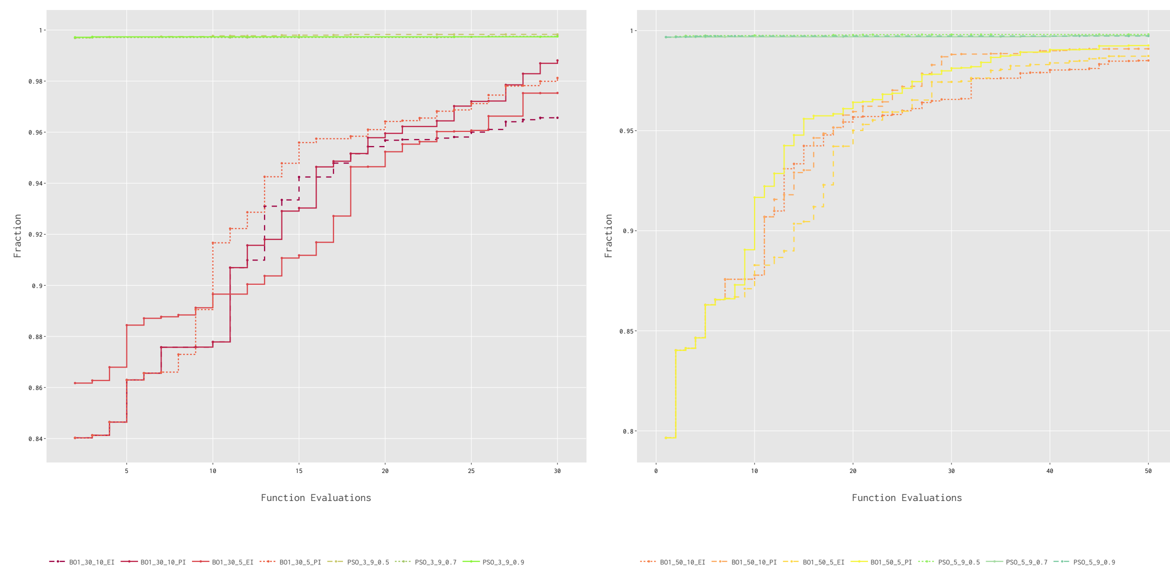

Figure 19: ECDF plots of the EAF for our optimization algorithms.

In Table 8, we summarize the top-3 Single-Objective algorithms for each scenario, those chosen based on optimal value (first two columns) and those we consider valid (last two columns). For the second category, there is not always more than one algorithm with feasibility. While PSO variants frequently achieve top objective values, their presence among valid solutions is limited, indicating feasibility challenges in some scenarios. Conversely, BO methods, particularly the 50-evaluation PI variant, consistently produce a higher number of valid optimal solutions, demonstrating a better balance between optimization quality and constraint satisfaction.
To interpret these results, we also generate Table 9, where we present the mean and standard deviation of each of the Single-Objective algorithms located in top-3 rankings. These are sorted in ascending order of mean based on non-valid optimal solutions. Concerning the valid ones, we also highlight how frequently each algorithm appeared among the top valid solutions for reliability purposes. PSO discovers only one optimal solution over all scenarios, with the 50-9-0.5 variant identifying none. If we rank the BO variations, 50-10-PI indicates the lowest mean with success in 10 scenarios. Some BO algorithms, such as 50-5-PI, exhibit higher standard deviation, suggesting that while they often find good solutions, their performance is more variable.
The evaluation of SAMO-COBRA configurations across all scenarios, as demonstrated in Table 10, reveals clear differences in optimization effectiveness as measured by Hypervolume (HV) and IGD+ metrics, given by *pymoo* package [4]. HV basically measures the size of the space covered by the Pareto front, showing how well an algorithm explores good trade-offs between objectives. A larger hypervolume means better coverage and diversity of solutions. IGD+

Table 8: Top-3 algorithms with their non-valid and valid optimal solutions per scenario for 30 and 50 evaluations

Sc	**Top-3 (30 ev)**	**Top-3 (50 ev)**	**Top-3 valid (30 ev)**	**Top-3 valid (50 ev)**
1	BO_30_5_EI (0.165) BO_30_5_PI (0.169) BO_30_10_PI (0.218)	PSO_5_9_0.9 (-0.183) PSO_5_9_0.5 (-0.168) PSO_5_9_0.7 (-0.095)	BO_30_5_EI (0.165) BO_30_5_PI (0.169) BO_30_10_PI (0.218)	BO_50_10_PI (0.135) BO_50_5_PI (0.157) BO_50_5_EI (0.165)
2	PSO_3_9_0.5 (-0.202) PSO_3_9_0.9 (-0.007) BO_30_5_PI (0.112)	PSO_5_9_0.5 (-0.258) PSO_5_9_0.7 (0.018) BO_50_5_PI (0.112)	BO_30_5_PI (0.112) BO_30_10_PI (7.525)	BO_50_5_PI (0.112) BO_50_10_PI (7.525)
3	PSO_3_9_0.7 (-0.075) BO_30_10_PI (2.944) BO_30_5_PI (3.517)	PSO_5_9_0.5 (-0.101) BO_50_5_EI (0.605) BO_50_10_EI (2.888)	BO_30_5_PI (11.470) BO_30_10_PI (19.023) BO_30_5_EI (30.876)	BO_50_5_PI (4.042) BO_50_10_PI (7.014) BO_50_5_EI (30.876)
4	PSO_3_9_0.5 (-0.256) PSO_3_9_0.7 (-0.086) BO_30_5_PI (0.425)	PSO_5_9_0.7 (-0.211) BO_50_5_PI (0.425) BO_50_10_PI (0.641)	BO_30_10_PI (1.581) BO_30_5_PI (3.377) BO_30_5_EI (15.672)	BO_50_10_PI (1.581) BO_50_5_PI (3.377) BO_50_10_EI (11.600)
5	PSO_3_9_0.9 (-0.170) PSO_3_9_0.7 (-0.087) PSO_3_9_0.5 (-0.030)	PSO_5_9_0.5 (-0.022) PSO_5_9_0.9 (-0.001) BO_50_10_PI (1.189)	BO_30_5_PI (37.713)	BO_50_5_EI (17.327) BO_50_5_PI (37.713)
6	PSO_3_9_0.5 (-0.147) PSO_3_9_0.7 (-0.130) PSO_3_9_0.9 (-0.079)	PSO_5_9_0.5 (0.000) BO_50_5_EI (0.605) BO_50_10_EI (2.888)	BO_30_5_PI (11.470) BO_30_10_PI (19.023) BO_30_5_EI (30.876)	BO_50_5_PI (4.042) BO_50_10_PI (7.014) BO_50_5_EI (30.876)
7	PSO_3_9_0.5 (-0.113) BO_30_10_PI (3.873) BO_30_5_PI (6.349)	PSO_5_9_0.5 (-0.110) BO_50_5_PI (0.792) BO_50_10_PI (2.543)	BO_30_5_PI (19.714) BO_30_10_PI (25.298)	BO_50_5_PI (2.998) BO_50_10_PI (8.174)
8	BO_30_10_PI (1.928) BO_30_10_EI (3.602) BO_30_5_EI (4.540)	BO_50_10_PI (0.079) BO_50_10_EI (0.789) BO_50_5_PI (0.856)	BO_30_10_EI (3.602) BO_30_5_EI (4.540)	BO_50_10_EI (0.789) BO_50_5_PI (0.856)
9	PSO_3_9_0.5 (-0.080) PSO_3_9_0.9 (0.201) BO_30_10_PI (1.518)	BO_50_10_PI (1.141) BO_50_10_EI (1.355) BO_50_5_PI (1.734)	BO_30_10_PI (1.518) PSO_3_9_0.9 (2.130) BO_30_5_PI (2.197)	PSO_5_9_0.7 (0.611) BO_50_10_PI (1.141) BO_50_10_EI (1.355)
10	PSO_3_9_0.5 (-0.285) BO_30_5_PI (-0.161) PSO_3_9_0.7 (-0.113)	PSO_5_9_0.5 (-0.170) BO_50_5_PI (-0.161) BO_50_10_PI (-0.134)	PSO_3_9_0.5 (-0.285) BO_30_5_PI (-0.161) PSO_3_9_0.7 (-0.113)	PSO_5_9_0.5 (-0.170) BO_50_5_PI (-0.161) BO_50_10_PI (-0.134)
11	BO_30_10_PI (0.873) BO_30_5_EI (1.222) BO_30_5_PI (3.498)	BO_50_10_PI (0.508) BO_50_5_EI (0.911) BO_50_10_EI (2.813)	BO_30_10_PI (0.873) BO_30_5_EI (1.222) BO_30_5_PI (3.498)	BO_50_10_PI (0.508) BO_50_5_EI (0.911) BO_50_10_EI (2.813)

Table 9: Statistics of valid and non-valid optimal solutions per algorithm

Non-valid solutions			Valid solutions		
Algorithm	Mean	Std	Mean	Std	Count
PSO_3_9_0.5	-0.016	0.211	-0.285	0.000	1
PSO_5_9_0.5	0.010	0.191	-0.170	0.000	1
PSO_3_9_0.7	0.083	0.174	-0.113	0.000	1
PSO_5_9_0.7	0.113	0.176	0.611	0.000	1
PSO_3_9_0.9	0.119	0.159	2.130	0.000	1
PSO_5_9_0.9	0.147	0.155	-	-	0
BO_50_5_PI	0.928	1.013	5.904	11.365	9
BO_50_10_PI	1.196	1.134	3.304	3.418	10
BO_30_10_PI	1.664	1.215	8.554	9.257	9
BO_50_5_EI	1.789	1.641	16.031	13.585	5
BO_50_10_EI	2.161	1.639	4.139	4.370	4
BO_30_5_PI	2.795	2.654	8.956	11.404	10
BO_30_5_EI	3.923	2.710	13.892	13.016	6
BO_30_10_EI	5.366	5.393	3.602	0.000	1

(Inverted Generational Distance Plus) measures how close the found solutions are to the true best Pareto front, by capturing both the distance and distribution quality of the solutions. A smaller IGD+ value means the solutions are closer and more evenly spread along the true front.

Notably, the 50-10 configuration consistently attains the highest HV values, indicating that it discovers lighting solutions that dominate a larger portion of the objective space, effectively balancing low energy consumption with high lighting uniformity. Concurrently, this configuration also achieves the lowest IGD+ scores, demonstrating that its approximate Pareto fronts closely match the true Pareto fronts, thus capturing a diverse and well-converged set of optimal trade-offs.

Conversely, configurations with fewer iterations or initial points, such as 30-5 and 30-10, generally exhibit lower HV and higher IGD+ values, suggesting a more limited exploration and less accurate approximation of the Pareto front. These differences highlight the benefit of increased computational budget and sampling in navigating the complex optimization landscape. Additionally, certain scenarios, such as Scenario 2, show overall higher IGD+ values across configurations, indicating increased optimization difficulty, likely due to environmental or sensor variability affecting solution quality. In contrast, scenarios where HV values are uniformly high and IGD+ values are low reflect more stable and well-defined optimization problems.

Overall, the metrics demonstrate that 50-10 is the most robust and effective SAMO-COBRA setting across varied scenarios, providing confidence that increasing iterations and initial points enhances both convergence quality and solution diversity in smart lighting optimization.

Table 11 presents the average elapsed time per algorithm, both including and excluding the DIALux UI manipulation. This distinction offers a clearer understanding of the actual computational time required by each algorithm, separate from the overhead introduced by the simulation interface. Assuming each DIALux simulation iteration takes approximately 3 minutes (i.e., 180 seconds), the expected total simulation time for a 30- and 50-budget run is 5,400 seconds (90 minutes) and 9,000 seconds (150 minutes), respectively. However, this is just an assumption as we do not actually have a standard average time over all evaluations.

Table 10: Metrics (Hypervolume and IGD+) of SAMO-COBRA throughout all scenarios

Sc.	Config.	HV	IGD+
1	30_5	66.786	0.043
	30_10	44.982	0.015
	50_5	134.484	0.033
	50_10	**166.562**	**0.003**
2	30_5	53.234	1.544
	30_10	44.982	1.515
	50_5	**253.931**	**0.012**
	50_10	160.945	1.499
3	30_5	122.730	0.055
	30_10	44.982	0.056
	50_5	**173.710**	0.017
	50_10	151.782	**0.014**
4	30_5	72.368	2.873
	30_10	50.638	2.846
	50_5	67.327	0.509
	50_10	**214.281**	**0.003**
5	30_5	122.730	1.043
	30_10	92.960	1.018
	50_5	**190.681**	**0.029**
	50_10	176.228	1.002
6	30_5	122.730	0.055
	30_10	44.982	0.056
	50_5	**173.710**	0.017
	50_10	151.782	**0.014**
7	30_5	122.730	0.039
	30_10	73.872	0.022
	50_5	125.642	0.037
	50_10	**139.941**	**0.001**
8	30_5	60.954	1.716
	30_10	44.982	1.702
	50_5	128.715	**0.036**
	50_10	**149.238**	1.644
9	30_5	68.878	1.031
	30_10	44.982	0.971
	50_5	**204.572**	**0.072**
	50_10	133.707	0.928
10	30_5	61.046	2.519
	30_10	44.982	2.501
	50_5	171.102	2.464
	50_10	**220.345**	**0.019**
11	30_5	106.680	1.500
	30_10	44.982	1.469
	50_5	149.398	1.470
	50_10	**178.628**	**0.008**

Table 11: Average elapsed time per algorithm, with and without DIALux simulation (approx.)

Algorithm	**Runtime Incl. DIALux**		**Runtime Excl. DIALux**	
	30 Budget	**50 Budget**	**30 Budget**	**50 Budget**
BO	5850.3 (≈1h 38m)	9661.5 (≈2h 41m)	450.3 (≈7.5m)	661.5 (≈11m)
SAMO-COBRA	6308.2 (≈1h 45m)	10573.0 (≈2h 56m)	908.2 (≈15m)	1573.0 (≈26m)
PSO	6382.2 (≈1h 47m)	11779.3 (≈3h 17m)	982.2 (≈16.5m)	2779.3 (≈46m)

7 Discussion

The findings of this thesis offer a comprehensive understanding of the comparative performance of optimization algorithms for intelligent lighting control in office environments. BO, particularly the variant using the Probability of Improvement acquisition function, consistently demonstrated robust performance across all scenarios. Configurations initialized with 5 points frequently outperformed those with ten, highlighting the advantage of preserving more of the evaluation budget for optimization rather than exploration. These observations were consistently supported across convergence curves and attainment plots, restating BO's ability to achieve feasibility early and consistently, even under constrained evaluation budgets. In contrast, PSO exhibited limited feasibility, with performance largely dependent on initialization and configuration tuning. SAMO-COBRA displayed more scenario-sensitive behavior, excelling under Multi-Objective formulations and when larger evaluation budgets were available.
The convergence analysis further supports these findings. BO exhibited a steady progression toward optimal solutions from the earliest iterations, often surpassing the warm-start baselines in both energy consumption and uniformity. The initial configurations, particularly those achieving uniformity levels above 0.7, offered a strong foundation for subsequent evaluations. Again, the 5 initialization samples allowed more evaluations to be spent refining promising regions of the search space, resulting in sharper early improvements and validating the observed sample-efficiency of these configurations. The interplay between warm-starting and targeted exploitation dynamics contributed substantially to BO's strong performance across scenarios. Expected Target Value (ETV) plots provided an additional lens into the convergence dynamics, reaffirming the previously established trends. PI-based BO configurations, especially with 5 initial samples, consistently achieved faster and more reliable convergence. While configurations with ten initial points occasionally reached marginally better final objective values under larger budgets, this came at the cost of slower early progress, illustrating the classic trade-off between exploration depth and evaluation efficiency. Conversely, EI-based configurations displayed delayed improvements, often concentrated near the end of the evaluation process. PSO remained suboptimal throughout, typically exhibiting poor convergence behavior and often failing to find feasible solutions. Collectively, these plots confirm the effectiveness of BO strategies that balance efficient initial sampling with strong exploitation mechanisms.
In the context of Multi-Objective optimization, the Pareto front analysis reinforces SAMO-COBRA's role as an effective solution for balancing energy consumption and lighting uniformity. The 50-10 configuration consistently produced well-distributed Pareto fronts with high Hypervolume and low IGD+ metrics, indicating effective trade-offs between objectives. These results illustrate that sufficient evaluation budgets are critical for high-quality Multi-Objective solutions. While 30-evaluation configurations generally underperformed, certain scenarios (e.g., 3, 7, and 9) demonstrated some competitive outcomes too.
The lumen distribution analysis provided practical insight into how each algorithm explored the solution space. BO exhibited adaptive behavior, often starting from high-lumen configurations and gradually reducing intensity in response to occupancy constraints. SAMO-COBRA maintained a broader exploration range, aligning with its Multi-Objective formulation, while PSO's unstructured behavior resulted in scattered and often infeasible outcomes. By analyzing these distribution patterns, targeted modifications were introduced to PSO's search space, such as narrowing or expanding intensity ranges per scenario. These adjustments led to significant improvements: in Scenarios 1 and 4, PSO became not only feasible but even outperformed BO whether with the presence or absence of warm-start. This suggests that algorithm performance

can be substantially enhanced through scenario-specific parameter tuning and informed search space design.

A holistic evaluation of all algorithms consolidates the earlier findings. BO with PI acquisition emerged as the most effective approach in Single-Objective settings, offering a reliable balance between feasibility, sample-efficiency, and convergence robustness. SAMO-COBRA excelled as Multi-Objective, particularly under the 50-10 configuration, which produced the most well-distributed Pareto fronts. Although PSO initially underperformed, its competitiveness improved significantly following targeted configuration adjustments, emphasizing the value of contextual adaptation. From a computational standpoint, BO provided the fastest execution times, while SAMO-COBRA, although slower, justified its computational cost with superior Pareto quality. These results underscore the complementary strengths of BO and SAMO-COBRA in addressing distinct optimization objectives within intelligent lighting systems.

Beyond algorithmic performance, the broader implications of this study are particularly relevant for energy-conscious and occupant-focused building environments. Minimizing energy consumption in lighting systems contributes directly to operational cost savings and significantly reduces the building's environmental footprint. Especially in commercial and office settings, where lighting constitutes a major component of energy use, even marginal gains in efficiency translate into substantial long-term savings. By adapting lighting output based on occupancy and ambient conditions, such systems support compliance with sustainability standards and contribute to greener building operations.

At the same time, maximizing lighting uniformity plays a vital role in ensuring occupant well-being. Uniform illumination reduces eye strain, enhances visual comfort, and supports higher levels of focus and task performance. This is especially critical in office environments where prolonged screen time and varying ambient conditions can affect employee comfort. Intelligent lighting systems that adapt illumination based on occupancy activity zones foster a more responsive and comfortable environment, leading to improved productivity, satisfaction, and overall health of building occupants. Therefore, the integration of smart lighting control represents not only an advancement in energy efficiency but also a foundation of human-centric building design.

7.1 Strengths

This thesis presents a robust and well-rounded approach to intelligent lighting control in office environments. First, the research employs a comprehensive evaluation framework, testing three distinct optimization algorithms: two BO variants and PSO across eleven randomized occupancy scenarios. This extensive experimentation simulates real-world variability in lighting needs, demonstrating the robustness and adaptability of each method under diverse conditions.

Second, a key strength of this thesis lies in the integration with DIALux evo, a professional-grade lighting simulation tool. By embedding the optimization process within a simulation environment, the study moves beyond theoretical analysis to provide realistic and practically applicable results. Evaluating lighting configurations in terms of energy consumption, desk-level illuminance, and lighting uniformity ensures that the proposed solutions meet both engineering performance metrics and human-centric comfort standards, as defined by EN 12464-1.

Another valuable contribution is the use of a multi-criteria evaluation strategy, emphasizing not only energy efficiency but also compliance with illuminance thresholds and uniformity across workspaces. The formulation of scenario-dependent feasibility, requiring 500 lux only on occupied desks, reflects a nuanced understanding of operational flexibility. While our system

applies consistent penalties across all scenarios, it allows practical tolerance for lighting on vacant desks, encouraging more energy-efficient yet context-aware solutions.
Moreover, this work emphasizes reproducibility, a hallmark of careful research. By explicitly setting random seeds, documenting initial sampling strategies, and detailing algorithm configurations, the experiments are fully transparent and replicable. This commitment to reproducibility increases the credibility and utility of the findings for future research.

7.2 Limitations

Despite the strengths of this work, several limitations should be acknowledged. One limitation of this study is that each algorithm was executed only once per configuration, without repeated runs using different random seeds. While such a process is recommended to account for the stochastic nature of Bayesian Optimization methods, this was not implemented due to time and computational constraints. As a result, the reported performance metrics may be influenced by specific random seeds, and conclusions drawn from single runs may not fully capture the variability or robustness of each algorithm and configuration.
Secondly, the entire evaluation process is based on a simulation environment that, while highly detailed and professional-grade, may not fully capture real-world factors such as reflections from furniture, sensor latency, or erratic occupant behavior, as the digital environment created does not accurately correspond to a real one. This reliance on simulation introduces a potential gap in generalizability, as the system's performance has not yet been validated in physical office environments. Moreover, the integration with DIALux evo relies on UI-based automation, which is both time-consuming and technically fragile. This limits the scalability of the experiments and poses challenges for large-scale reproducibility, particularly when testing on more complex building layouts.
In addition, the occupancy scenarios used throughout the experiments are based on randomized but static cases. While this allows for a controlled and repeatable evaluation, it does not capture temporal changes such as people entering or leaving rooms during the day. Consequently, the system lacks real-time adaptability and cannot respond to dynamic occupancy or ambient lighting conditions, which would be essential for deploying a closed-loop smart lighting solution in practice.
Furthermore, daylight is omitted from the simulation setup. Although this exclusion simplifies the optimization problem and isolates the effects of artificial lighting, it also limits the ecological validity of the results. In office environments where daylight plays a crucial role, especially near windows or skylights, the omission may overestimate the need for artificial lighting and thereby distort the energy-performance trade-offs of certain configurations.
Finally, the use of warm-starts in Bayesian Optimization and SAMO-COBRA introduces a potential bias. While initializing the optimization with pre-sampled configurations improves convergence speed, it may also limit the algorithm's exploration of the search space, increasing the risk of converging prematurely to suboptimal regions. This bias, combined with the above limitations, outlines the current boundaries of the proposed framework and points toward directions for future improvement. The bias also exists when minimizing the search space to enhance PSO, as it removes a big part of the accepted values for a lamp, making it unable to choose lower lumen values which in reality are in the range of lumens. This may also increase energy as higher lumen levels are depicted.

8 Conclusion

This work introduces a black-box approach to guide smart indoor lighting, aiming to balance energy use with occupant comfort in commercial offices. The challenge was tackled through both Single-Objective and Multi-Objective models, all kept within the illuminance and uniformity limits laid out by EN 12464-1. Because each run depends on the DIALux evo engine and the output is non-differentiable, classic gradient-based search tools proved ineffective. As a result, we tested three search strategies: a simple Bayesian Optimization, the surrogate-based SAMO-COBRA, and standard Particle Swarm Optimization.

To answer the primary research question, the results demonstrate that both single- and Multi-Objective warm-started Bayesian Optimization methods outperform the commonly used baseline PSO algorithm in optimizing indoor lighting systems. BO achieved lower energy consumption while maintaining adequate uniformity and comfort, and did so more consistently across scenarios. In contrast, PSO often failed to meet feasibility constraints and was less reliable in complex occupancy-driven settings. These findings confirm the need for a higher budget concerning PSO, and also the ability of BO methods to succeed in a limited budget.

Extensive experiments across eleven different occupancy scenarios showed that BO, especially when warm-started and using the Probability of Improvement acquisition function, reliably produced feasible and high-quality lighting configurations. BO exhibited better convergence behavior and sample efficiency compared to PSO, which often struggled to find feasible solutions due to not having a warm-start and being sensitive to hyperparameter settings. SAMO-COBRA, although more demanding in terms of computation, delivered strong performance in Multi-Objective settings, particularly when there was enough evaluation budget available.

Key findings include the effectiveness of BO in discovering configurations that minimize energy consumption while maintaining adequate uniformity and desk lux levels. Moreover, the study underscored the importance of incorporating occupancy information into the optimization pipeline to reduce over-illumination of vacant spaces and improve overall system efficiency.

After the intervention of optimizing PSO, we can observe that for scenarios where uniformity should exceed the predefined threshold, utilizing the distribution of lumens might be insightful. However, this applies only when the distribution is more informative and it is clear where the lumen values are located, i.e., which regions are better. Furthermore, such a process of reducing the search space may not be suggested as optimal, as it consists of bias of the optimization procedures towards higher values of lumen and thus energy, ending up with suboptimal solutions.

In conclusion, this work validates the applicability of black-box optimization methods for adaptive lighting control in indoor environments. By leveraging occupancy-based sensor data, the proposed system demonstrated the ability to deliver energy-conscious configurations without compromising user comfort. These findings contribute valuable insights to the growing field of smart building technologies and offer a practical foundation for real-world deployment.

Future work

This thesis provides a solid foundation for intelligent lighting control via black-box optimization. Future research can build on this work by constructing real-time optimization systems that respond dynamically to changing occupancy and daylight conditions. Lighting setups could be constantly adjusted throughout the day by connecting the optimization loop with real-time sensor data, providing greater responsiveness and efficiency in operational environments.

Another interesting avenue is to personalize lighting setups based on individual user preferences or behavioral patterns. For example, some people may favor brighter conditions, whereas others are more sensitive to glare. Integrating user profile or preference learning mechanisms can result in a more user-centric control system that not only saves energy but also improves comfort and pleasure.
The optimization framework might be modified to encompass different building subsystems, such as HVAC or controlled blinds, allowing for cross-domain optimization. Joint control of lighting and climate systems has the potential to greatly increase building energy efficiency. Furthermore, future implementations could investigate how this approach expands to bigger or multi-room office settings.
Sensor integration is another area that has room for growth. Incorporating more environmental inputs, such as CO_2 levels, ambient lighting, or computer vision-based occupancy recognition, can increase the accuracy of control triggers and optimization decisions.
Finally, future initiatives may benefit from quickening the optimization process. Creating surrogate models to approximate DIALux simulation outputs would enable speedier evaluations and accommodate larger-scale or real-time applications. This acceleration could also aid integration with current Building Management Systems (BMS), allowing a modular version of the framework to be included in commercial smart building platforms, transitioning the system from experimental to deployable.

References

[1] Y. Al horr, M. Arif, M. Katafygiotou, A. Mazroei, A. Kaushik, and E. Elsarrag. Impact of indoor environmental quality on occupant well-being and comfort: A review of the literature. *International Journal of Sustainable Built Environment*, 5(1):1–11, 2016.

[2] O. Ayan and B. Turkay. Iot-based energy efficiency in smart homes by smart lighting solutions. In *2020 21st International Symposium on Electrical Apparatus Technologies (SIELA)*, pages 1–5, 2020.

[3] C. Benavente-Peces. On the energy efficiency in the next generation of smart buildings—supporting technologies and techniques. *Energies*, 12(22), 2019.

[4] J. Blank and K. Deb. pymoo: Multi-objective optimization in python. *IEEE Access*, 8:89497–89509, 2020.

[5] I. Chew, D. Karunatilaka, C. P. Tan, and V. Kalavally. Smart lighting: The way forward? reviewing the past to shape the future. *Energy and Buildings*, 149:180–191, 2017.

[6] G. Cimini, A. Freddi, G. Ippoliti, A. Monteriù, and P. Matteo. A smart lighting system for visual comfort and energy savings in industrial and domestic use. *Electric Power Components and Systems*, 43, 2015.

[7] A. Colmenar-Santos, L. N. Terán de Lober, D. Borge-Diez, and M. Castro-Gil. Solutions to reduce energy consumption in the management of large buildings. *Energy and Buildings*, 56:66–77, 2013.

[8] R. de Winter, B. van Stein, and T. Bäck. Samo-cobra: A fast surrogate assisted constrained multi-objective optimization algorithm. In H. Ishibuchi, Q. Zhang, R. Cheng, K. Li, H. Li, H. Wang, and A. Zhou, editors, *Evolutionary Multi-Criterion Optimization*, pages 270–282, Cham, 2021. Springer International Publishing.

[9] C. Doerr, H. Wang, F. Ye, S. van Rijn, and T. Bäck. IOHprofiler: A Benchmarking and Profiling Tool for Iterative Optimization Heuristics. *arXiv e-prints:1810.05281*, 2018.

[10] European Committee for Standardization. EN 12464-1:2021 – Light and lighting – Lighting of work places – Part 1: Indoor work places, 2021.

[11] P. I. Frazier. A tutorial on bayesian optimization, 2018.

[12] D. GmbH. Dialux evo (version 13 patch 4), 2024.

[13] A. K. Gopalakrishna, T. Özçelebi, A. Liotta, and J. J. Lukkien. Exploiting machine learning for intelligent room lighting applications. In *2012 6th IEEE International Conference Intelligent Systems*, pages 406–411, 2012.

[14] T. Head, MechCoder, G. Louppe, M. Pak, N. Bouraqadi, and S. Gelly. Scikit-optimize: Sequential model-based optimization with a Scipy-based api. *GitHub repository*, 2018.

[15] X. Lin, Q. Guo, D. Yuan, and M. Gao. Bayesian optimization framework for hvac system control. *Buildings*, 13(2):314, 2023.

[16] E.-N. D. Madias, P. A. Kontaxis, and F. V. Topalis. Application of multi-objective genetic algorithms to interior lighting optimization. *Energy and Buildings*, 125:66–74, 2016.

[17] P. Mandal, D. Dey, and B. Roy. Indoor lighting optimization: a comparative study between grid search optimization and particle swarm optimization. *Journal of Optics*, 48:429–441, 2019.

[18] L. Martirano. A smart lighting control to save energy. pages 132–138, 2011.

[19] J. Miranda. Pyswarms: a research toolkit for particle swarm optimization in python, 2018.

[20] M. H. Ngo, X. V. C. Nguyen, Q. K. Duong, and H. S. Nguyen. Adaptive smart lighting control based on genetic algorithm. In *2019 25th Asia-Pacific Conference on Communications (APCC)*, pages 320–325, 2019.

[21] A. M. Omer. Energy, environment and sustainable development. *Renewable and Sustainable Energy Reviews*, 12(9):2265–2300, 2008.

[22] A. G. Putrada, M. Abdurohman, D. Perdana, and H. H. Nuha. Machine learning methods in smart lighting toward achieving user comfort: A survey. *IEEE Access*, 10:45137–45178, 2022.

[23] U. S. Shanthamallu, A. Spanias, C. Tepedelenlioglu, and M. Stanley. A brief survey of machine learning methods and their sensor and iot applications. In *2017 8th International Conference on Information, Intelligence, Systems Applications (IISA)*, pages 1–8, 2017.

[24] M. Soheilian, G. Fischl, and M. Aries. Smart lighting application for energy saving and user well-being in the residential environment. *Sustainability*, 13(11), 2021.

[25] K. R. Wagiman, M. N. Abdullah, M. Y. Hassan, and N. H. Mohammad Radzi. A new metric for optimal visual comfort and energy efficiency of building lighting system considering daylight using multi-objective particle swarm optimization. *Journal of Building Engineering*, 43:102525, 2021.

[26] W. Xu, B. Svetozarevic, L. D. Natale, P. Heer, and C. N. Jones. Data-driven adaptive building thermal controller tuning with constraints: A primal–dual contextual bayesian optimization approach. *Applied Energy*, 358:122493, 2024.

[27] S. Yelisetti, R. Kumar, V. Gupta, A. Saxena, and R. Lamba. Modelling and analysis of home energy management system using intelligent algorithms. pages 1–6, 2020.

[28] S. Yelisetti, V. K. Saini, R. Kumar, and R. Lamba. Energy consumption cost benefits through smart home energy management in residential buildings: An indian case study. In *2022 IEEE IAS Global Conference on Emerging Technologies (GlobConET)*, pages 930–935, 2022.

[29] D. Zhai, S. Bahadure, B. Ong, and Y. Soh. Conserving energy of office lighting by smart features in singapore buildings. pages 1–5, 2024.

[30] R. Zhang, C. Campanella, S. Aristizabal, A. Jamrozik, J. Zhao, P. Porter, S. Ly, and B. A. Bauer. Impacts of dynamic led lighting on the well-being and experience of office occupants. *International Journal of Environmental Research and Public Health*, 17(19), 2020.

[31] Y. Zhang, Y. Zhong, Y. Gong, and L. Zheng. The optimization of visual comfort and energy consumption induced by natural light based on pso. *Sustainability*, 11(1), 2019.

Appendices

Tables for Scenarios 2-11 are provided below.

Table 12: Feasible solutions for Scenario 2

Algorithm	Settings	Energy	Uniformity	Desk Lux per surface
BO	30_10_PI	510.8	0.73	[630, 637, 633, 678, 679, 666]
		503.5	0.72	[634, 637, 629, 661, 661, 646]
		467.5	0.67	[542, 573, 613, 650, 629, 612]
	30_5_PI	439.4	0.73	[516, 481, 513, 538, 610, 631]
	50_10_PI	510.8	0.73	[630, 637, 633, 678, 679, 666]
		503.5	0.72	[634, 637, 629, 661, 661, 646]
		467.5	0.67	[542, 573, 613, 650, 629, 612]
	50_5_PI	439.4*	0.73*	[516, 481, 513, 538, 610, 631]*
		437.6	0.71	[553, 551, 582, 574, 593, 531]
SAMO-COBRA	50_10	470.7	0.73	[599, 603, 587, 653, 595, 568]
	50_5	476.9	**0.78**	[613, 641, 622, 598, 576, 571]
		455.9	0.75	[565, 592, 602, 601, 574, 523]

Table 13: Feasible solutions for Scenario 3

Algorithm	Settings	Energy	Uniformity	Desk Lux per surface
BO	30_10_PI	476.8	0.69	[560, 599, 589, 616, 651, 645]
	30_5_EI	519.9	0.72	[648, 655, 651, 688, 683, 669]
	30_5_PI	513.1	0.73	[647, 647, 644, 676, 663, 653]
		449.0	0.69	[540, 524, 589, 568, 592, 643]
	50_10_PI	476.8	0.69	[560, 599, 589, 616, 651, 645]
		409.2	0.67	[512, 556, 513, 511, 594, 543]
	50_5_EI	519.9	0.72	[648, 655, 651, 688, 683, 669]
	50_5_PI	513.1	0.73	[647, 647, 644, 676, 663, 653]
		449.0	0.69	[540, 524, 589, 568, 592, 643]
		396.1*	0.57*	[519, 392, 480, 529, 554, 613]*
		392.6**	0.61**	[505, 387, 539, 506, 540, 542]**
SAMO-COBRA	50_10	444.0	0.65	[599, 560, 556, 530, 571, 592]
	50_5	492.8	**0.74**	[615, 647, 627, 654, 619, 613]

Table 14: Feasible solutions for Scenario 4

Algorithm	Settings	Energy	Uniformity	Desk Lux per surface
BO	30_10_PI	465.3**	0.71**	[623, 627, 627, 611, 573, 514]**
	30_5_EI	512.7	0.71	[647, 654, 633, 686, 680, 654]
	30_5_PI	420.2*	0.58*	[568, 558, 511, 585, 584, 493]*
		501.4	**0.72**	[624, 636, 631, 665, 657, 637]
		483.2	0.71	[604, 622, 605, 649, 641, 603]
		437.3	0.66	[561, 577, 531, 593, 586, 532]
	50_10_EI	**428.4**	0.62	[590, 513, 599, 533, 531, 614]
	50_10_PI	465.3	0.71	[623, 627, 627, 611, 573, 514]
	50_5_EI	512.7	0.71	[647, 654, 633, 686, 680, 654]
	50_5_PI	420.2*	0.58*	[568, 558, 511, 585, 584, 493]*
		501.4	**0.72**	[624, 636, 631, 665, 657, 637]
		483.2	0.71	[604, 622, 605, 649, 641, 603]
		437.3	0.66	[561, 577, 531, 593, 586, 532]
SAMO-COBRA	50_10	433.8	0.70	[533, 532, 555, 577, 573, 542]

Table 15: Feasible solutions for Scenario 5

Algorithm	Settings	Energy	Uniformity	Desk Lux per surface
BO	30_5_PI	487.1	0.72	[602, 612, 610, 643, 636, 627]
	50_5_EI	452.9**	0.64**	[635, 639, 552, 599, 532, 465]**
	50_5_PI	487.1	0.72	[602, 612, 610, 643, 636, 627]
	50_10_PI	360.0*	0.50*	[561, 442, 540, 557, 341, 374]*
SAMO-COBRA	30_5	**451.6**	0.70	[615, 602, 642, 578, 491, 518]
	50_10	473.7	**0.72**	[546, 581, 618, 637, 627, 599]
	50_5	**451.6**	0.70	[615, 602, 642, 578, 491, 518]

Table 16: Feasible solutions for Scenario 6

Algorithm	Settings	Energy	Uniformity	Desk Lux per surface
BO	30_10_PI	476.8	0.69	[560, 599, 589, 616, 651, 645]
	30_5_EI	519.9	0.72	[648, 655, 651, 688, 683, 669]
	30_5_PI	513.1	0.73	[647, 647, 644, 676, 663, 653]
		449.0	0.69	[540, 524, 589, 568, 592, 643]
	50_10_PI	476.8	0.69	[560, 599, 589, 616, 651, 645]
		409.2	0.67	[512, 556, 513, 511, 594, 543]
	50_5_EI	519.9	0.72	[648, 655, 651, 688, 683, 669]
	50_5_PI	513.1	0.73	[647, 647, 644, 676, 663, 653]
		449.0	0.69	[540, 524, 589, 568, 592, 643]
		396.1*	0.57*	[519, 392, 480, 529, 554, 613]*
		392.6**	0.61**	[505, 387, 539, 506, 540, 542]**
SAMO-COBRA	50_10	444.0	0.65	[599, 560, 556, 530, 571, 592]
	50_5	492.8	**0.74**	[615, 647, 627, 654, 619, 613]
		462.7	0.68	[592, 586, 596, 587, 602, 599]

Table 17: Feasible solutions for Scenario 7

Algorithm	**Settings**	**Energy**	**Uniformity**	**Desk Lux per surface**
BO	30_10_PI	489.9	**0.74**	[574, 590, 615, 644, 661, 654]
	30_5_PI	496.9	**0.74**	[618, 619, 635, 655, 646, 643]
		487.9	0.72	[619, 632, 642, 554, 563, 656]
		479.9	0.67	[590, 614, 604, 645, 642, 595]
	50_10_PI	489.9	**0.74**	[574, 590, 615, 644, 661, 654]
		440.6	0.67	[555, 516, 564, 519, 564, 596]
	50_5_PI	496.9	**0.74**	[618, 619, 635, 655, 646, 643]
		487.9	0.72	[619, 632, 642, 554, 563, 656]
		479.9	0.67	[590, 614, 604, 645, 642, 595]
		416.5	0.65	[532, 514, 521, 548, 535, 566]
		402.9*	0.58*	[538, 495, 508, 543, 504, 522]*
		395.7**	0.66**	[506, 529, 506, 511, 485, 535]**
SAMO-COBRA	30_5	451.6	0.70	[615, 602, 642, 578, 491, 518]
	50_10	434.2	0.69	[522, 594, 591, 512, 487, 553]
	50_5	451.6	0.70	[615, 602, 642, 578, 491, 518]

Table 18: Feasible solutions for Scenario 8

Algorithm	**Settings**	**Energy**	**Uniformity**	**Desk Lux per surface**
BO	30_5_EI	519.1	0.71	[636, 653, 658, 687, 682, 669]
	50_5_EI	519.1	0.71	[636, 653, 658, 687, 682, 669]
	30_5_PI	504.3	0.74	[623, 638, 634, 678, 672, 640]
	50_5_PI	504.3	0.74	[623, 638, 634, 678, 672, 640]
	50_10_PI	396.5*	0.68*	[480, 566, 541, 570, 433, 378]*
SAMO-COBRA	50_5	446.1	0.73	[531, 520, 555, 526, 503, 502]
	50_10	453.5	0.69	[534, 564, 525, 584, 658, 578]
		450.8	0.68	[509, 539, 624, 649, 611, 541]

Table 19: Feasible solutions for Scenario 9

Algorithm	**Settings**	**Energy**	**Uniformity**	**Desk Lux per surface**
BO	50_10_PI	**334.7***	0.50*	[471, 440, 550, 524, 315, 300]*
PSO	3_9_0.5	271.4*	0.60*	[362, 352, 329, 256, 297, 437]*
	5_9_0.7	345.0**	0.55**	[456, 471, 579, 506, 300, 288]**
SAMO-COBRA	30_5	429.7	0.26	[484, 584, 562, 567, 564, 594]
	50_5	429.7	0.26	[484, 584, 562, 567, 564, 594]
		467.6	0.69	[586, 565, 594, 565, 587, 600]
		476.9	0.77	[632, 601, 568, 583, 589, 607]
	50_10	452.4	0.72	[547, 530, 599, 577, 549, 593]
		460.6	0.73	[519, 564, 575, 622, 593, 604]

Table 20: Feasible solutions for Scenario 10

Algorithm	Settings	Energy	Uniformity	Desk Lux per surface
BO	30_5_EI	116.0	0.07	[408, 405, 74.7, 64.5, 35.3, 42.6]
		150.8	0.28	[205, 216, 97, 229, 300, 199]
		154.0	0.29	[274, 322, 72.3, 155, 133, 80.6]
		244.4	0.35	[301, 321, 182, 316, 328, 339]
		245.3	0.57	[287, 348, 379, 262, 256, 307]
	30_10_EI	124.7	0.25	[166, 249, 56.6, 59.9, 62.3, 199]
		188.3	0.32	[278, 386, 156, 164, 121, 360]
		189.2	0.45	[203, 237, 240, 204, 219, 266]
		272.9	0.57	[341, 315, 371, 429, 414, 286]
	50_5_EI	**104.8**	0.13	[44, 136, 178, 200, 152, 264]
		150.8	0.28	[205, 216, 97, 229, 300, 199]
		154.0	0.29	[274, 322, 72.3, 155, 133, 80.6]
		219.3	0.33	[296, 341, 175, 176, 295, 389]
		229.5	0.34	[430, 439, 166, 226, 191, 255]
		230.4	0.44	[401, 421, 235, 285, 213, 272]
		245.3	0.57	[287, 348, 379, 262, 256, 307]
	50_10_EI	110.9	0.11	[28.6, 71.1, 211, 157, 227, 199]
		124.7	0.25	[166, 249, 56.6, 59.9, 62.3, 199]
		152.7	0.32	[182, 314, 165, 122, 149, 227]
		189.2	0.45	[203, 237, 240, 204, 219, 266]
		272.9	0.57	[341, 315, 371, 429, 414, 286]
	30_5_PI	165.9	0.29	[318, 335, 202, 143, 105, 184]
		166.8	0.48	[271, 281, 213, 140, 124, 219]
	30_10_PI	187.3	0.32	[340, 250, 221, 345, 285, 245]
		232.3	0.33	[243, 247, 210, 329, 415, 289]
		232.9	0.41	[373, 325, 294, 318, 282, 248]
		250.9	0.44	[471, 362, 314, 385, 278, 236]
		265.4	0.47	[342, 253, 351, 444, 316, 340]
		271.6	0.49	[407, 317, 406, 487, 332, 250]
		272.9	0.57	[341, 315, 371, 429, 414, 286]
	50_5_PI	123.1	0.29	[98.7, 175, 88.2, 167, 141, 215]
		166.7	0.32	[238, 251, 202, 251, 185, 132]
		166.8*	0.48*	[271, 281, 213, 140, 124, 219]*
		249.3	0.52	[375, 329, 277, 324, 353, 348]
	50_10_PI	169.0	0.22	[453, 325, 133, 133, 70, 84]
		187.3	0.32	[340, 250, 221, 345, 285, 245]
		189.2	0.45	[203, 237, 240, 204, 219, 266]
		221.2	0.49	[232, 181, 409, 248, 218, 339]
		238.3	0.59	[268, 281, 222, 382, 394, 238]

Table 21: Feasible solutions for Scenario 10 (continue)

Algorithm	Settings	Energy	Uniformity	Desk Lux per surface
PSO	5_9_0.9	294.9	0.62	[373, 343, 338, 323, 427, 434]
		309.9	0.66	[470, 401, 358, 289, 381, 391]
		322.4	0.69	[447, 387, 387, 307, 399, 437]
	3_9_0.5	190.4*	0.65*	[266, 229, 184, 214, 296, 328]*
		249.3	0.66	[275, 266, 292, 310, 388, 415]
	5_9_0.5	253.5	0.61	[399, 351, 334, 412, 321, 274]
		257.2	0.63	[374, 325, 355, 428, 339, 272]
		259.2	0.65	[400, 352, 345, 422, 333, 277]
		260.4	0.66	[398, 353, 346, 427, 336, 273]
		261.3	0.67	[399, 353, 347, 430, 337, 274]
	3_9_0.9	324.8	0.61	[465, 469, 372, 398, 463, 454]
SAMO-COBRA	30_5	49.6	0.29	[99, 40.4, 92.7, 114, 81.2, 80]
		77.8	0.33	[84.6, 63.5, 159, 200, 145, 126]
		101.6	0.37	[161, 82.5, 120, 184, 200, 170]
		118.5	0.39	[91.7, 98.8, 184, 268, 263, 215]
		271.6	0.43	[341, 312, 346, 409, 355, 379]
		275.8	0.47	[313, 280, 334, 306, 301, 485]
		289.1	0.52	[348, 261, 484, 395, 456, 359]
	30_10	49.6	0.29	[99, 40.4, 92.7, 114, 81.2, 80]
		77.8	0.33	[84.6, 63.5, 159, 200, 145, 126]
		101.6	0.37	[161, 82.5, 120, 184, 200, 170]
		118.5	0.39	[91.7, 98.8, 184, 268, 263, 215]
		140.6	0.41	[179, 116, 242, 231, 215, 250]
		142.8	0.46	[146, 122, 199, 212, 268, 292]
		164	0.49	[237, 151, 266, 287, 204, 224]
		191.2	0.53	[211, 177, 253, 319, 318, 311]
	50_5	49.6	0.29	[99, 40.4, 92.7, 114, 81.2, 80]
		77.8	0.33	[84.6, 63.5, 159, 200, 145, 126]
		101.6	0.37	[161, 82.5, 120, 184, 200, 170]
		115.9	0.49	[155, 85.2, 133, 177, 145, 216]
		289.1	0.52	[348, 261, 484, 395, 456, 359]
		322.1	0.53	[399, 452, 378, 460, 394, 408]
		384.4	0.71	[495, 503, 508, 516, 484, 478]
		409.9	0.73	[494, 456, 527, 517, 524, 562]
		479.5	0.74	[574, 625, 598, 596, 595, 634]
	50_10	8.0	0.16	[5.19, 4.51, 21.6, 24.2, 15, 33]
		49.6	0.29	[99, 40.4, 92.7, 114, 81.2, 80]
		31.0	0.36	[401, 324, 473, 486, 493, 402]
		95.5	0.48	[124, 86, 136, 155, 148, 163]
		164.0	0.49	[237, 151, 266, 287, 204, 224]
		191.2	0.53	[211, 177, 253, 319, 318, 311]
		365.9	0.58	[506, 499, 393, 324, 457, 618]
		432.1	0.67	[564, 549, 541, 478, 534, 638]
		458.2	0.70	[558, 567, 558, 566, 622, 653]

Table 22: Feasible solutions for Scenario 11

Algorithm	**Settings**	**Energy**	**Uniformity**	**Desk Lux per surface**
BO	50_10_PI	189.2*	0.45*	[203, 237, 240, 204, 219, 266]*
		348.6**	0.56**	[557, 416, 513, 477, 307, 325]**
SAMO-COBRA	30_5	317.1	0.37	[512, 412, 525, 342, 217, 305]
		444.4	0.60	[538, 552, 568, 603, 554, 597]
	50_5	317.1	0.37	[512, 412, 525, 342, 217, 305]
		444.4	0.60	[538, 552, 568, 603, 554, 597]
		453.0	0.76	[541, 573, 562, 595, 568, 590]
	50_10	427.1	0.61	[548, 489, 575, 543, 559, 572]

Convergence over iterations for Scenarios 2-11.

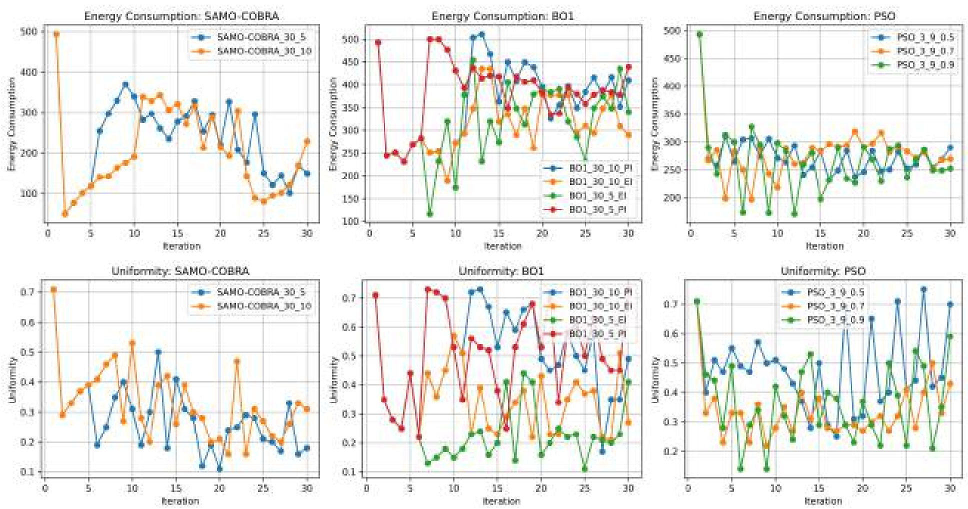

Figure 20: Energy and uniformity for Scenario 2 with 30-budget

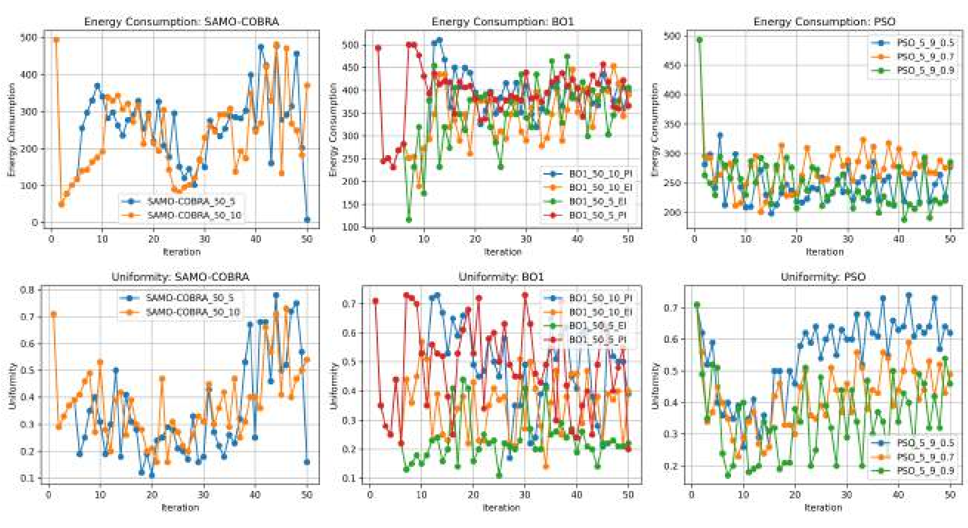

Figure 21: Energy and uniformity for Scenario 2 with 50-budget

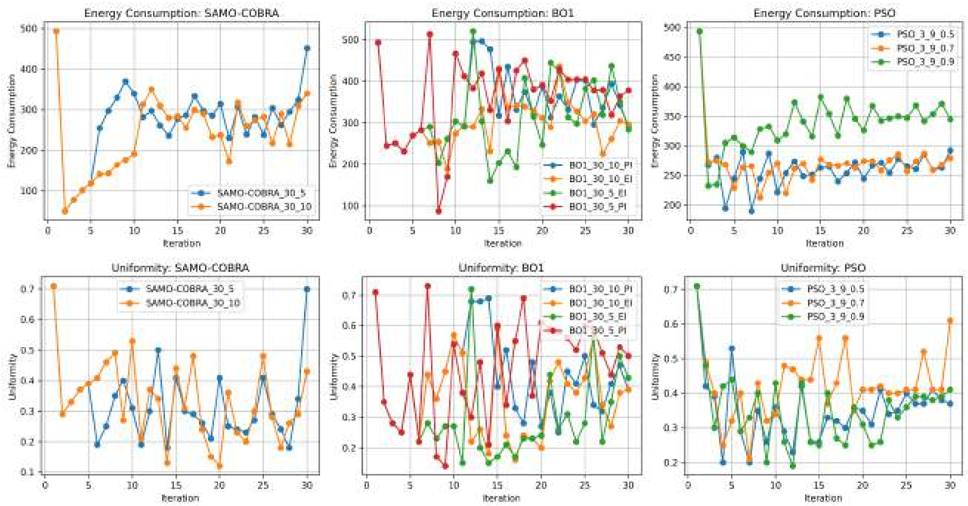

Figure 22: Energy and uniformity for Scenario 3 with 30-budget

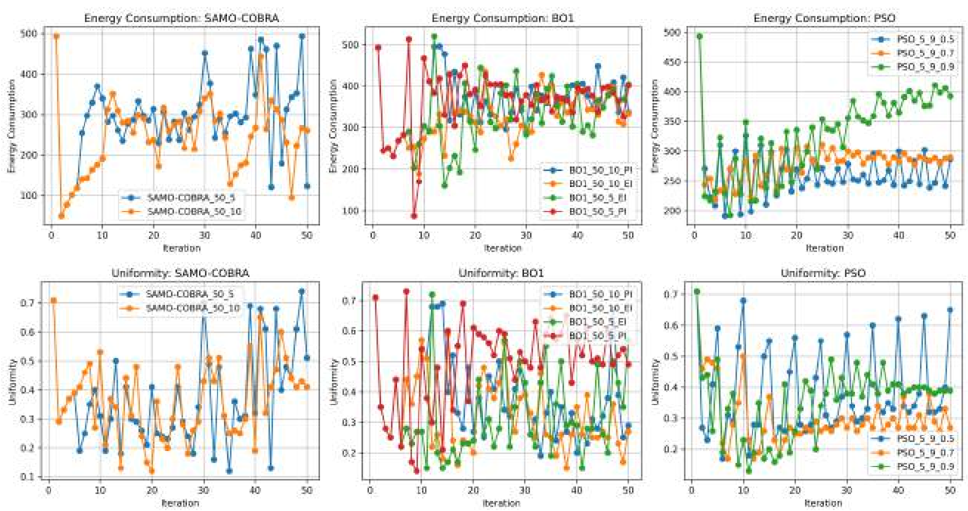

Figure 23: Energy and uniformity for Scenario 3 with 50-budget

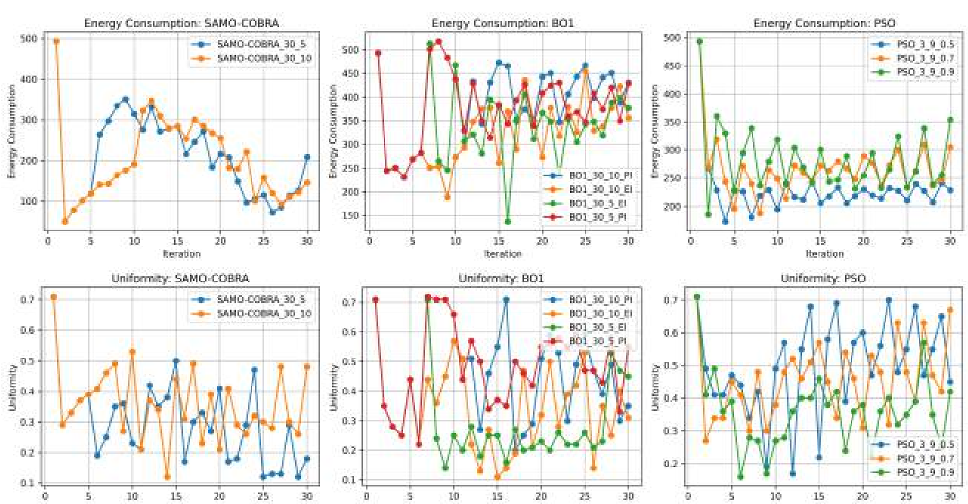

Figure 24: Energy and uniformity for Scenario 4 with 30-budget

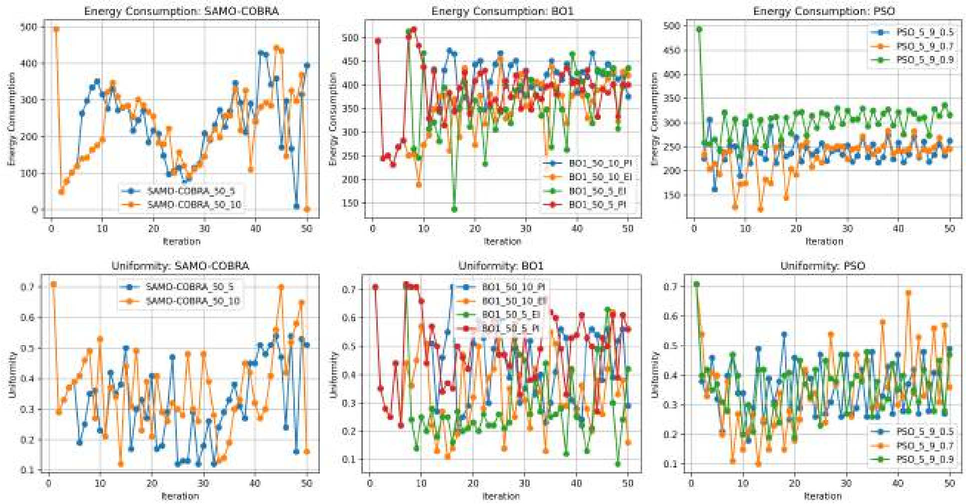

Figure 25: Energy and uniformity for Scenario 4 with 50-budget

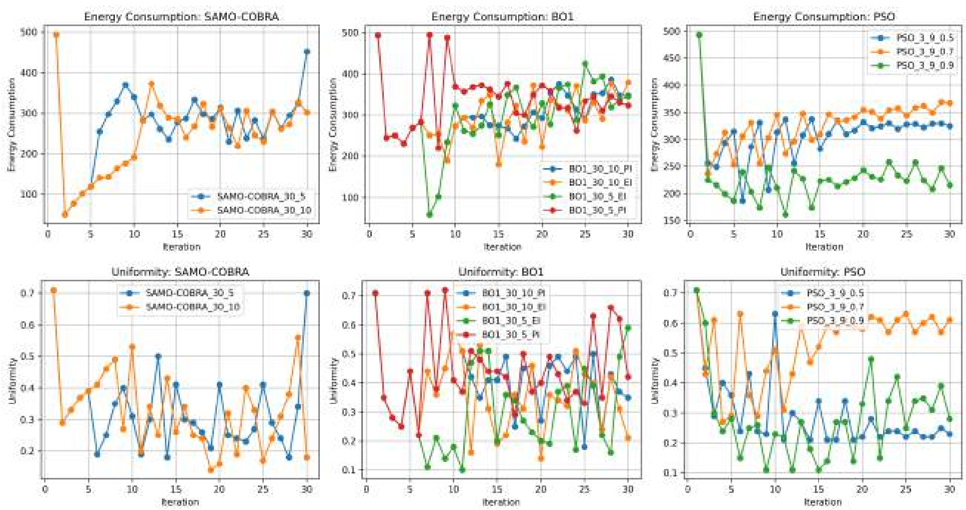

Figure 26: Energy and uniformity for Scenario 5 with 30-budget

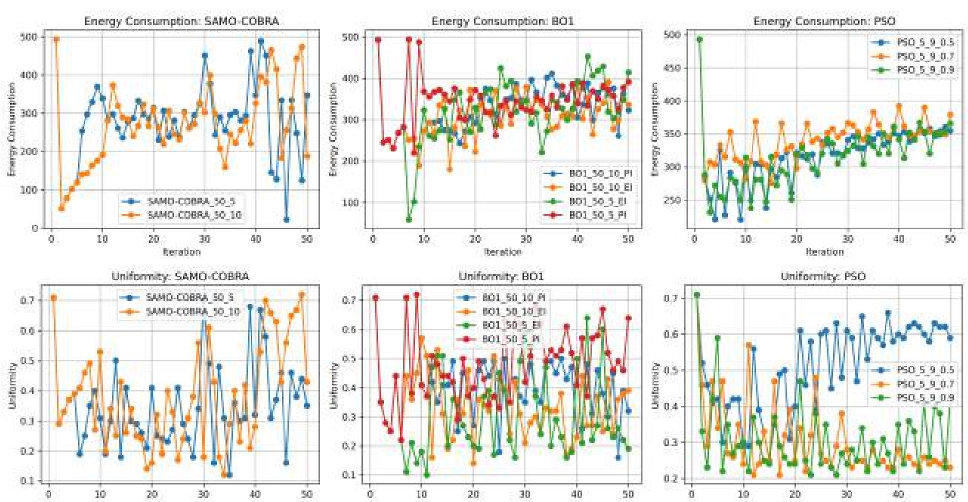

Figure 27: Energy and uniformity for Scenario 5 with 50-budget

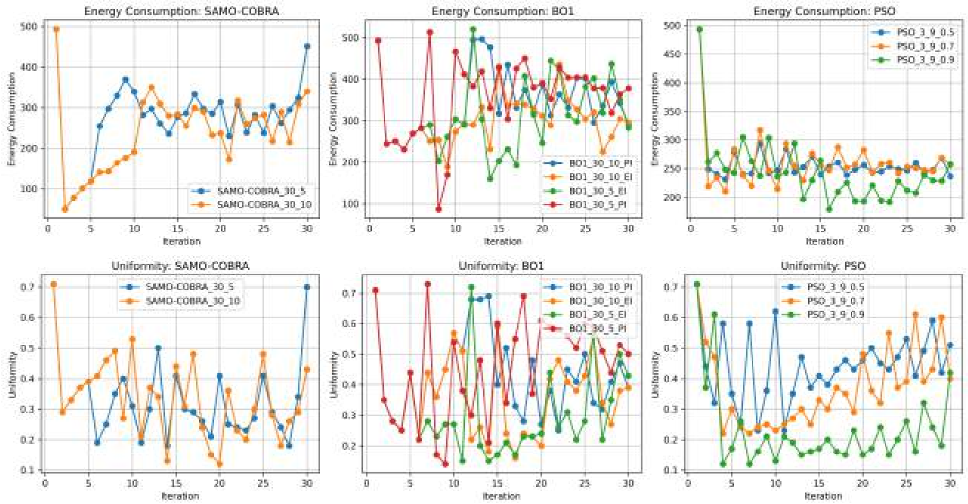

Figure 28: Energy and uniformity for Scenario 6 with 30-budget

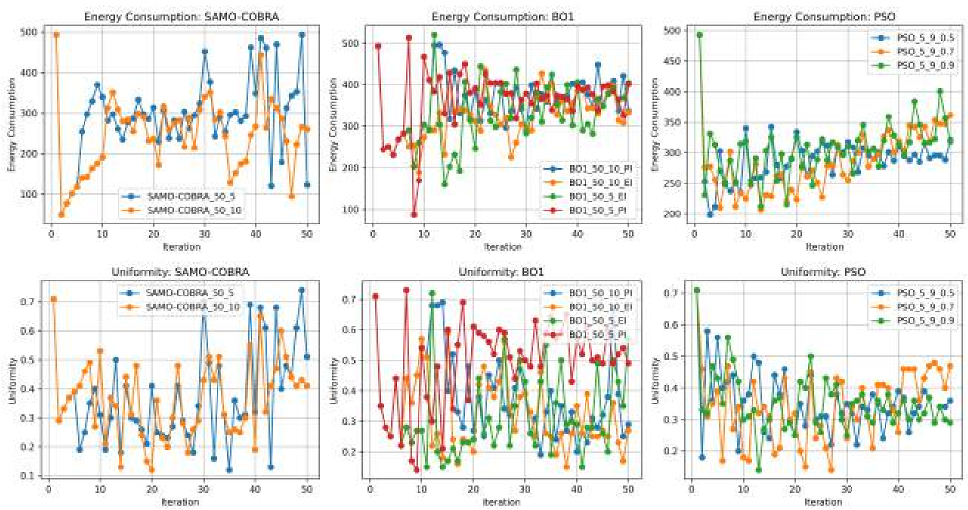

Figure 29: Energy and uniformity for Scenario 6 with 50-budget

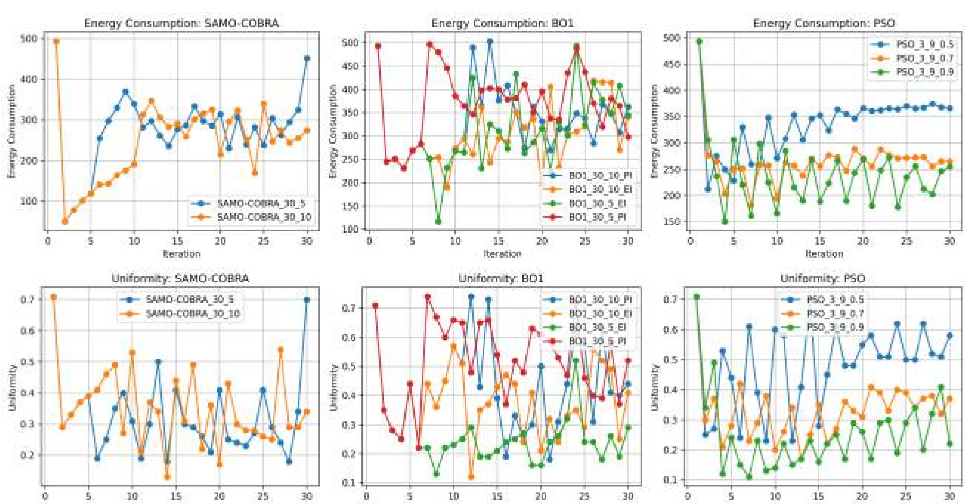

Figure 30: Energy and uniformity for Scenario 7 with 30-budget

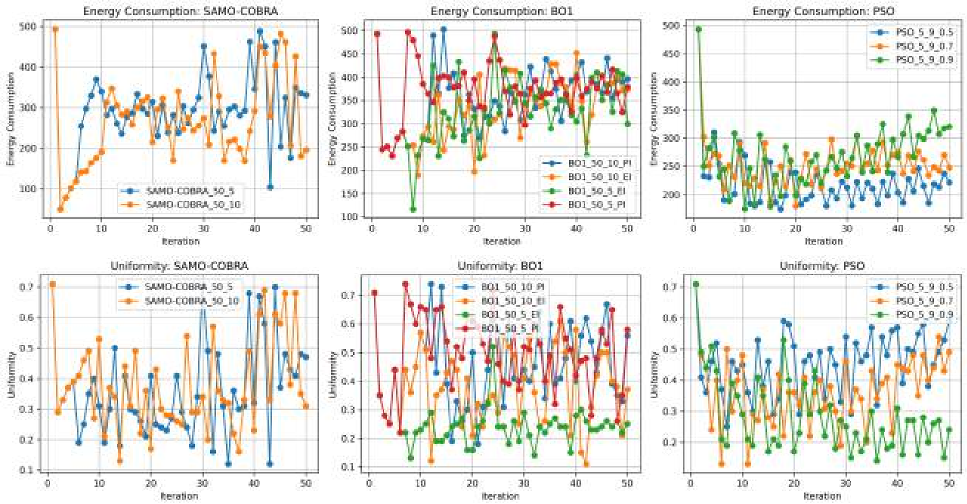

Figure 31: Energy and uniformity for Scenario 7 with 50-budget

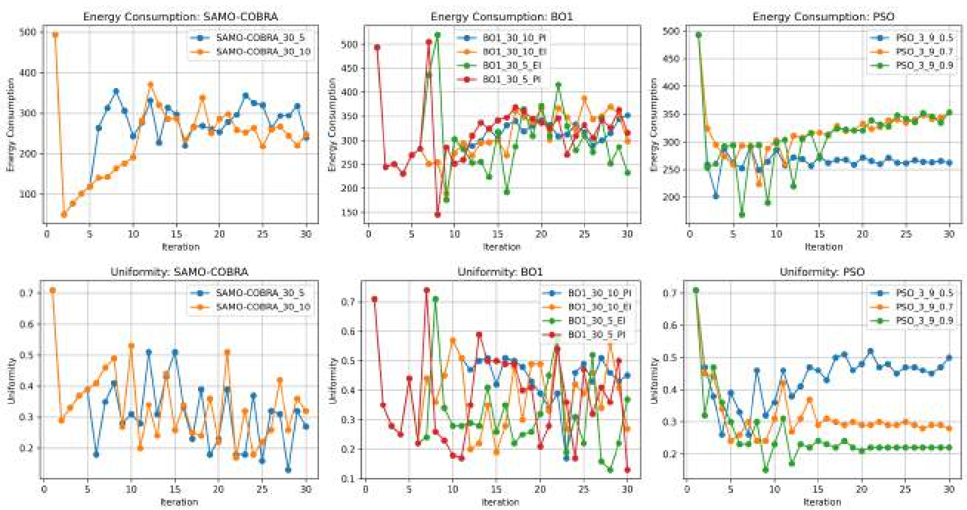

Figure 32: Energy and uniformity for Scenario 8 with 30-budget

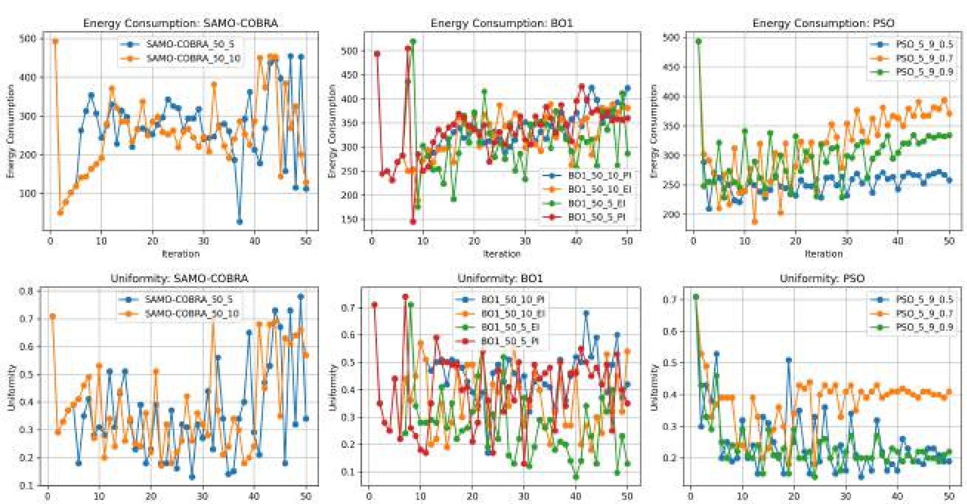

Figure 33: Energy and uniformity for Scenario 8 with 50-budget

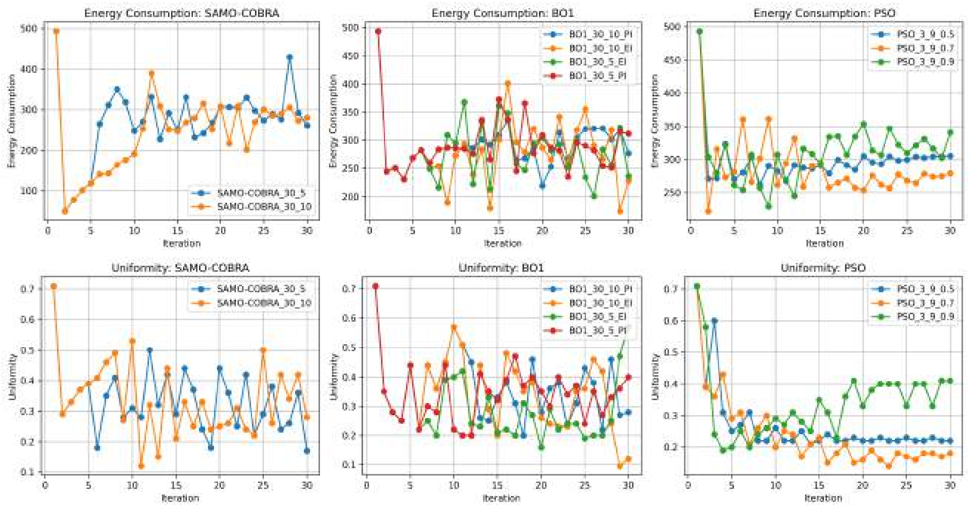

Figure 34: Energy and uniformity for Scenario 9 with 30-budget

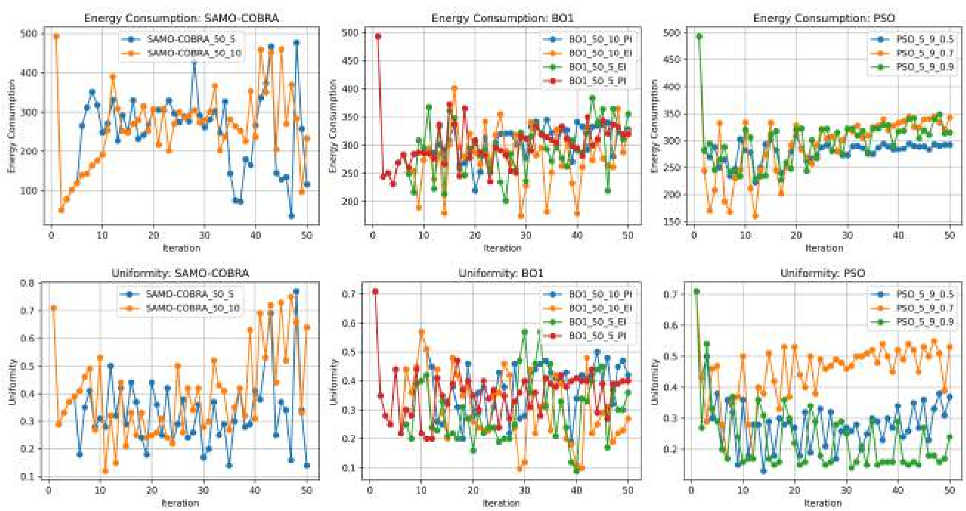

Figure 35: Energy and uniformity for Scenario 9 with 50-budget

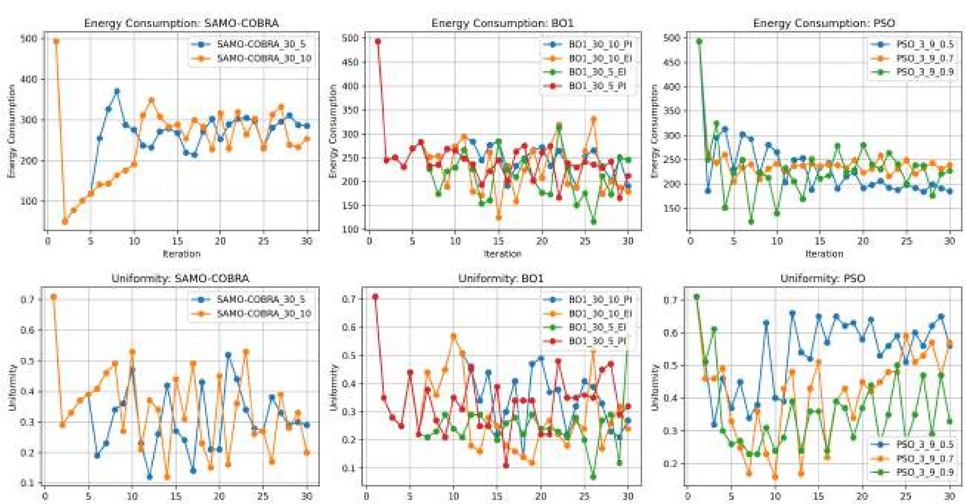

Figure 36: Energy and uniformity for Scenario 10 with 30-budget

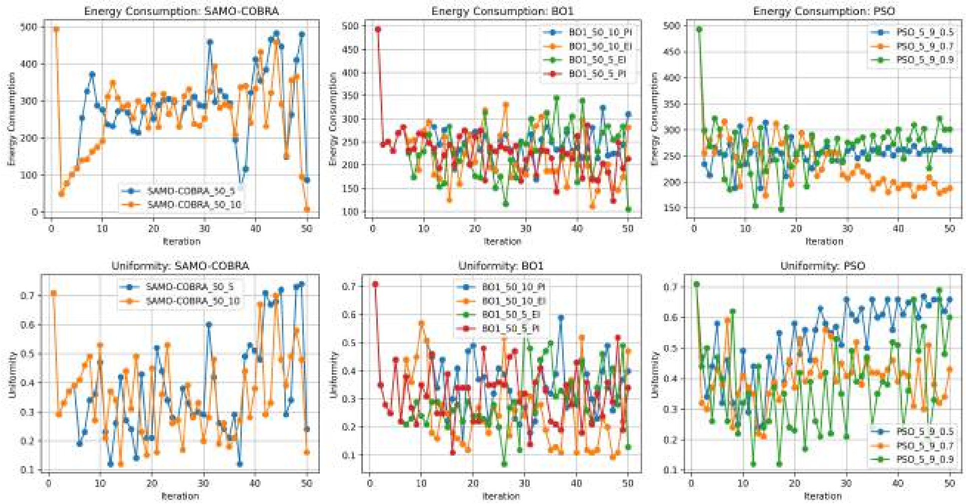

Figure 37: Energy and uniformity for Scenario 10 with 50-budget

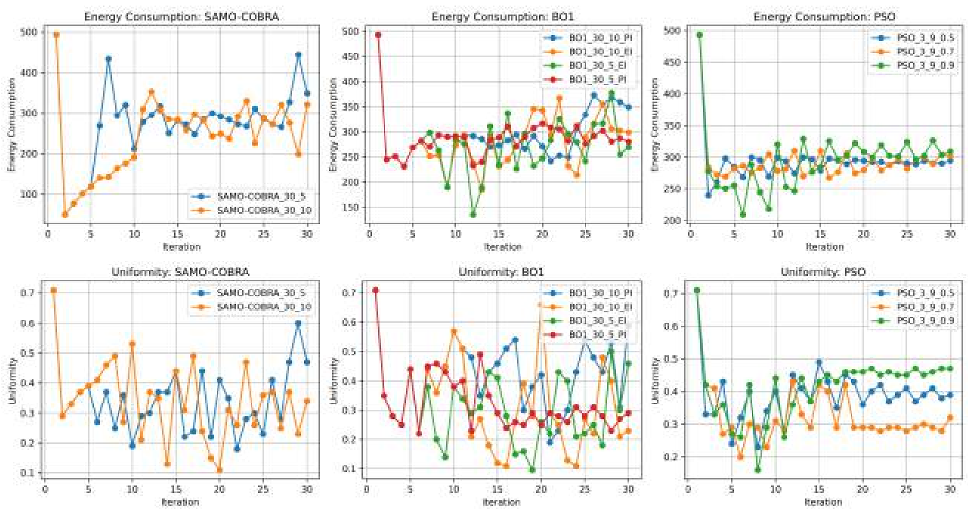

Figure 38: Energy and uniformity for Scenario 11 with 30-budget

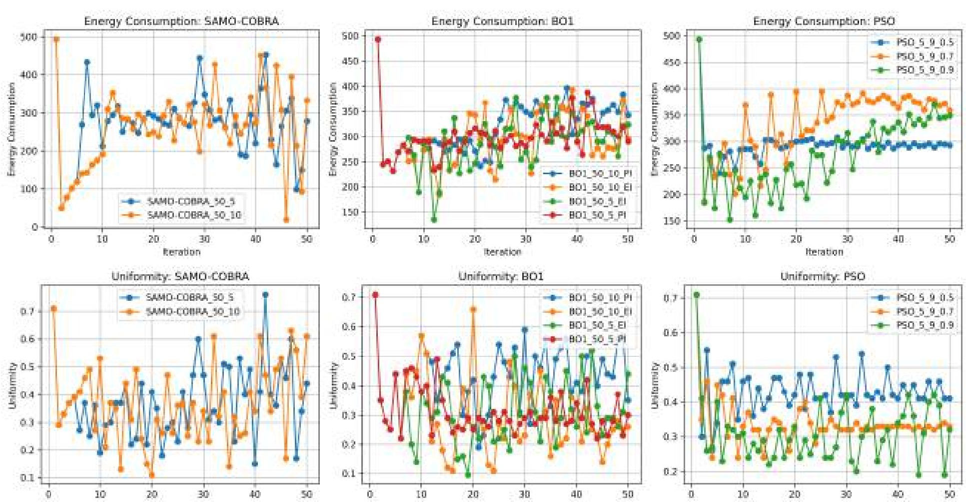

Figure 39: Energy and uniformity for Scenario 11 with 50-budget

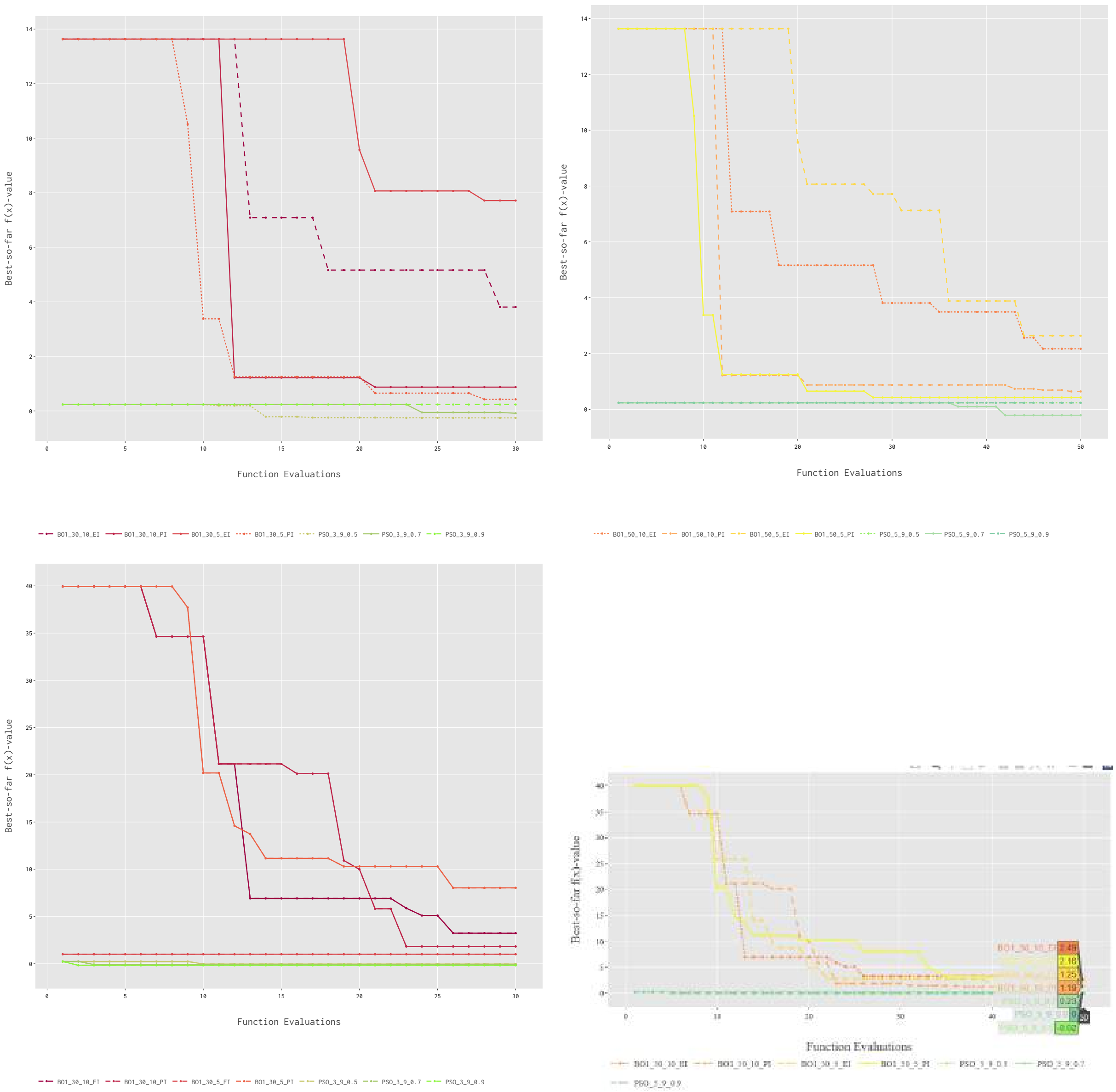

Figure 40: Function evaluations of Scenarios 4-5 for both 30- and 50- budget

Function evaluations plots for Scenarios 4-11.

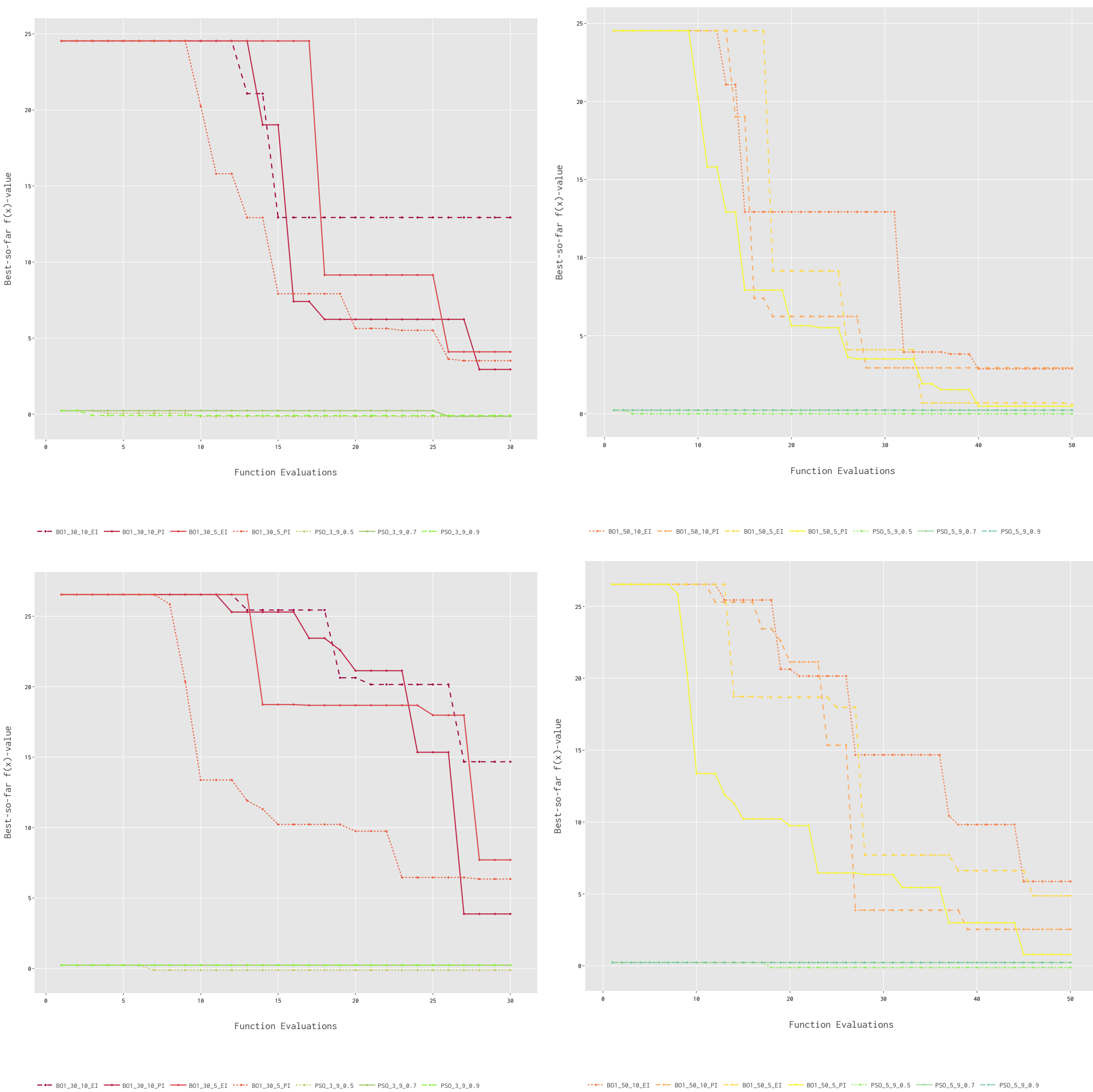

Figure 41: Function evaluations of Scenarios 6-7 for both 30- and 50- budget

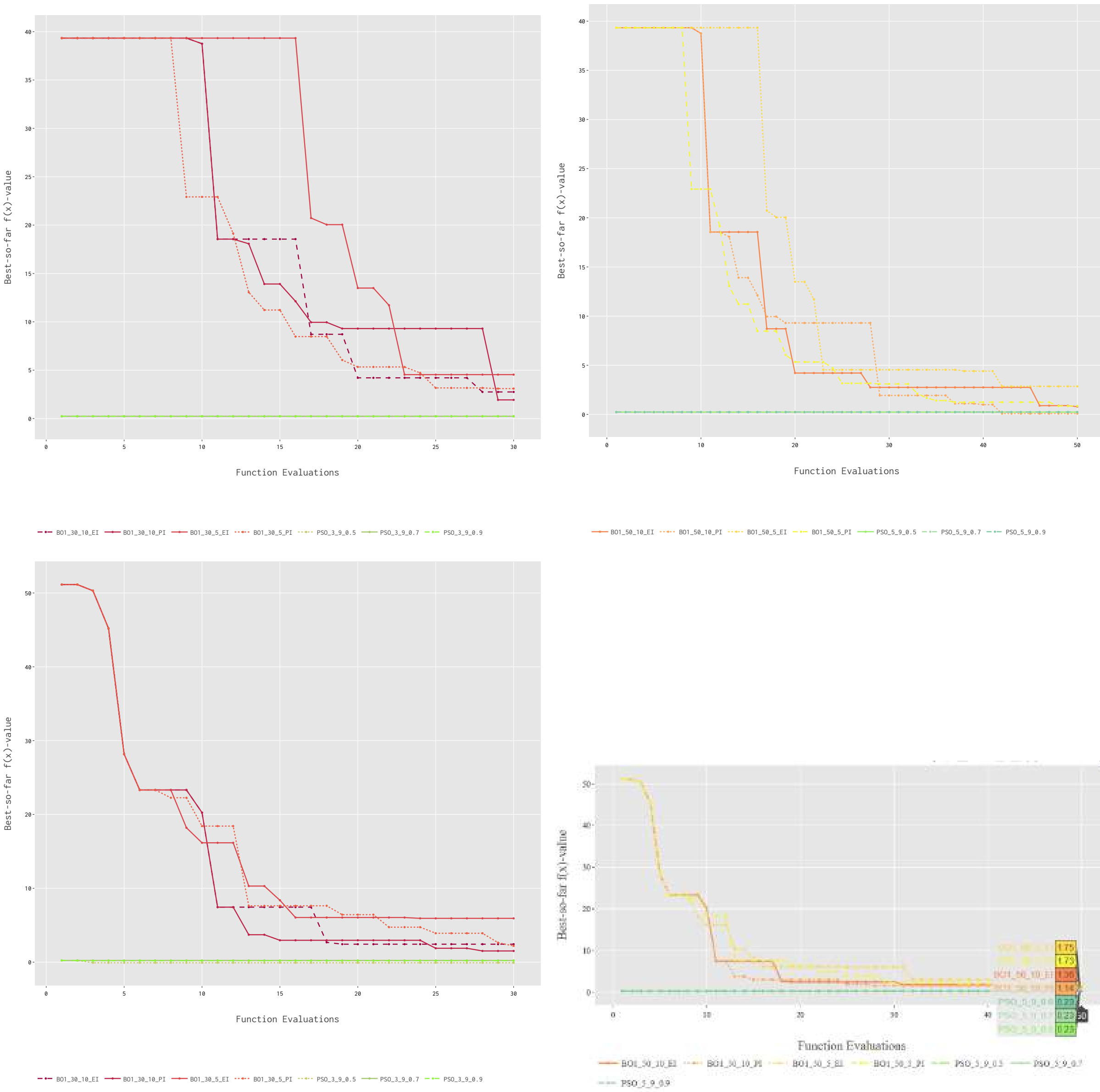

Figure 42: Function evaluations of Scenarios 8-9 for both 30- and 50- budget

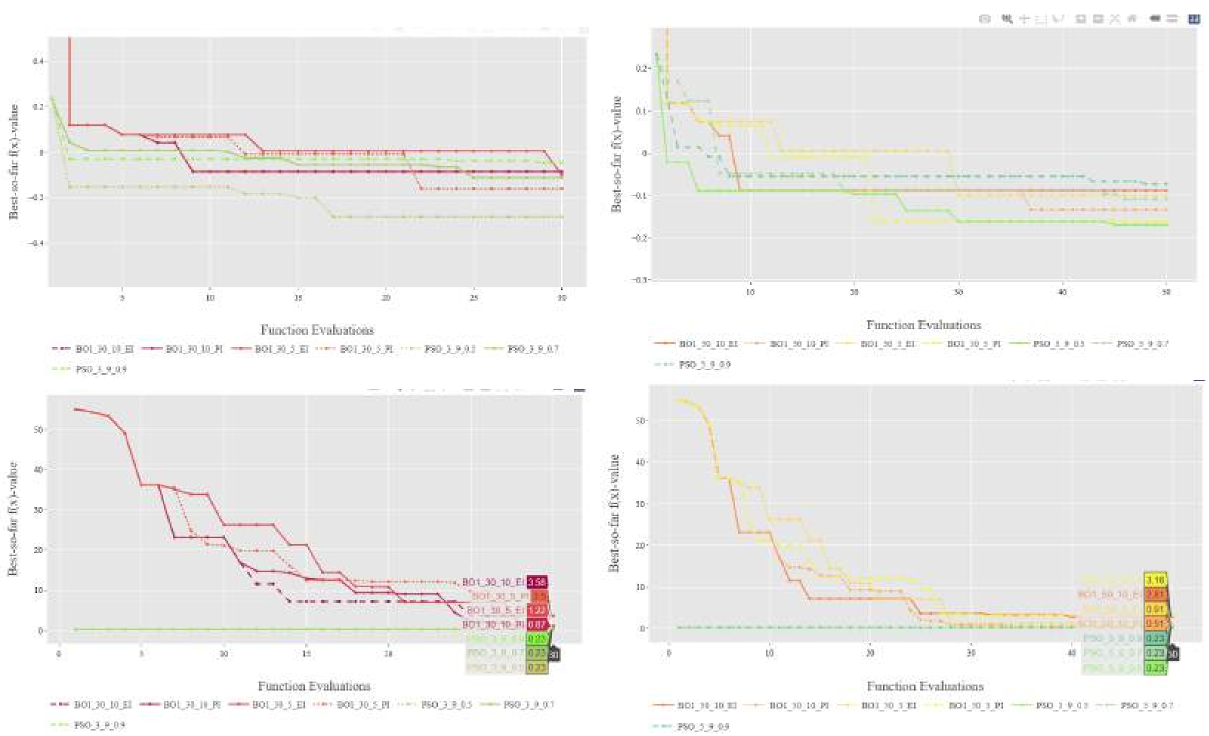

Figure 43: Function evaluations of Scenarios 10-11 for both 30- and 50- budget